1.3

IBM Backup and Restore Manager for z/VM User Guide

SC18-9523-15

Note:

Before using this information and the product it supports, read the "Notices" topic at the end of this information.

This edition applies to Version 1 Release 3 of IBM Backup and Restore Manager for z/VM (product number 5697-J06) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this information

IBM Backup and Restore Manager for z/VM® (also referred to as Backup and Restore Manager) is a tool to back up and restore CMS and non-CMS data in a VM environment.

This information provides instructions for operating IBM Backup and Restore Manager for z/VM. It is designed to help general users and system administrators use Backup and Restore Manager to backup and restore data and assumes a working knowledge of:

- z/VM operating system
- CMS (including XEDIT)
- SFS concepts and facilities
- REXX

In this book, ECKD is referred to as "CKD."

Service updates and support information

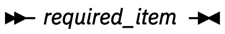
To find service updates and support information, including software Fix Packs, PTFs, Frequently Asked Questions (FAQs), technical notes, troubleshooting information, and downloads, refer to the Web page:

http://www.ibm.com/software/products/en/backup-restore-manager-for-zvm

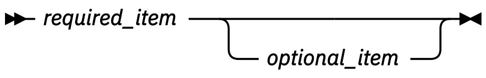
How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

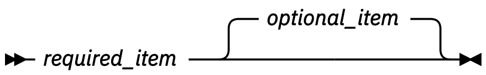
- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>--- symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >--- symbol indicates that a syntax diagram is continued from the previous line.
  - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).

- Optional items appear below the main path.

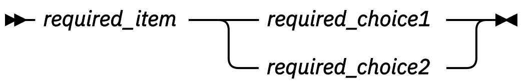
  

  If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

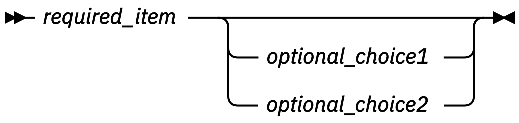
  

- If you can choose from two or more items, they appear vertically, in a stack.

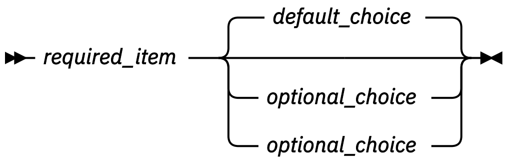
  If you *must* choose one of the items, one item of the stack appears on the main path.

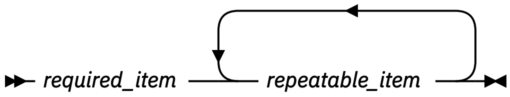
If choosing one of the items is optional, the entire stack appears below the main path.



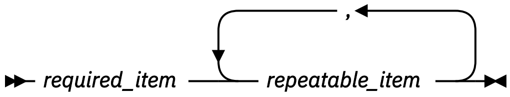
If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



- An arrow returning to the left, above the main line, indicates an item that can be repeated.



If the repeat arrow contains a comma, you must separate repeated items with a comma.



A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, *column-name*). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses; for example, (1).

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this documentation or any other Backup and Restore Manager documentation, use either of the following options:

- If you have questions or comments regarding z/VM publications and product documentation, please visit: http://www.vm.ibm.com/forms/http://www.vm.ibm.com/forms/
- Send your comments by email to zvmtools@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of IBM Backup and Restore Manager for z/VM, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

Chapter 1. Overview

These topics introduce the functionality and benefits provided by Backup and Restore Manager for z/VM.

What's new in IBM Backup and Restore Manager for z/VM

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

SC18-9523-15 for APAR PH42721, PTF UI78527

New function code changes associated with the command, **BKRBRT**, included a change that provides greater flexibility in the CMS file name requirements for the Batch Restore Job creation. For more information, see "Batch restore processing" on page 17 for more information.

SC18-9523-14

A new command, **BKRBRT**, has been added to allow you to generate batch-mode job statements to restore data owned by an individual user or MDISKs from a specific real DASD volume. For more information, see "BKRBRT command (Bulk Restore Tool)" on page 19 for more information.

SC18-9523-13

The **BKRUSER** command syntax was updated. For more information, see "BKRUSER" on page 32.

SC18-9523-12

The **RESTORE** command syntax was updated to enable an additional target location type (DEV). For more information, see "RESTORE command" on page 11.

SC18-9523-11

The BKRLIST description was updated. For more information, see "BKRLIST" on page 29.

SC18-9523-10

- Support for Byte File System (BFS) objects was added. For information about restoring BFS data, see "Restoring data from a catalog browser interface" on page 6.
- Information about performing a BFS file space restore was added. For more information, see "Performing a BFS file space restore" on page 10.
- The **RESTORE** command syntax was updated to support BFS File System objects. For more information, see "RESTORE command" on page 11.

What does Backup and Restore Manager do?

With Backup and Restore Manager you can back up and restore CMS and non-CMS data in a VM environment. You can create backup copies of your original data, and in the event of a disaster or other data loss, quickly restore your data to help ensure maximum availability.

System administrators regularly perform backups to help protect the system from loss of data and to help increase productivity and minimize downtime if a data loss occurs. In most cases, the administrator only retains a limited number of backup versions (for example, as a new backup is created, the administrator might delete the oldest version). This practice allows the administrator to save storage space; and retain backup copies from which the system can be restored, if needed.

Users can perform backups of data regularly to guard against data loss. Performing backups in this manner helps protect against accidental file deletion or file corruption.

Features

You can perform the following tasks using Backup and Restore Manager.

- Back up and restore CMS and non-CMS data (one file, a group of files, or an entire minidisk).
- Back up data to disk or tape.
- Back up data from a command-line interface.
- Perform batch restore processing.
- Restore data from a full panel catalog browser interface or a command-line interface, which you can also call from your REXX applications.

Supported data types

With Backup and Restore Manager you can back up and restore the following types of data.

- CMS formatted minidisk
- CMS SFS (Shared File System)
- Raw image dumps of CKD (Count Key Data) including z/OS®, Linux® on zSeries, and VSE volumes
- Raw image dumps of FBA (Fixed-Block Architecture) DASD devices

Supported storage media

Backup and Restore Manager supports the following types of storage media.

- IBM® 3480, 3490, and 3590 tape storage media
- CMS files on minidisk on in Shared File System (SFS)

IBM Tape Manager for z/VM is recommended when choosing tape, automated tape libraries, or virtual tape servers.

Terminology

Before you use Backup and Restore Manager, you should become familiar with these terms and concepts:

Backup catalog
The backup catalog collects and organizes metadata produced during the backup process.

Backup stream
The backup stream files consist of a combination of data that is contained in the backup and metadata that contains information about the task which created the backup stream.

Client
A *client* is a virtual machine (typically, a CMS user). Clients initiate requests through the primary backup service virtual machine.

Container
A *container* is an object that holds data (for example: a CMS minidisk, an SFS file space, or a CKD DASD extent). A container can hold zero (0) or more files.

Granule
A *granule* refers to metadata packages that are subsets of the backup stream. Each granule is a file that contains the results of a single backup process and includes information such as the job name and instance, job owner, date and time of job instantiation, and so on. Backup catalog content is generated using granules.

Accessing help

Access Backup and Restore Manager help in the following ways.

- To access the help menu, type:

  ```
  HELP ABKR
  ```

- To access help for a specific command, type HELP followed by the command name. For example:

  ```
  HELP ABKR RESTORE
  ```

Backup and Restore Manager processing

Backup and Restore Manager uses a combination of service virtual machines to accomplish backup and restore processing.

Figure 1 on page 3 shows how Backup and Restore Manager processes backup and user restore requests.

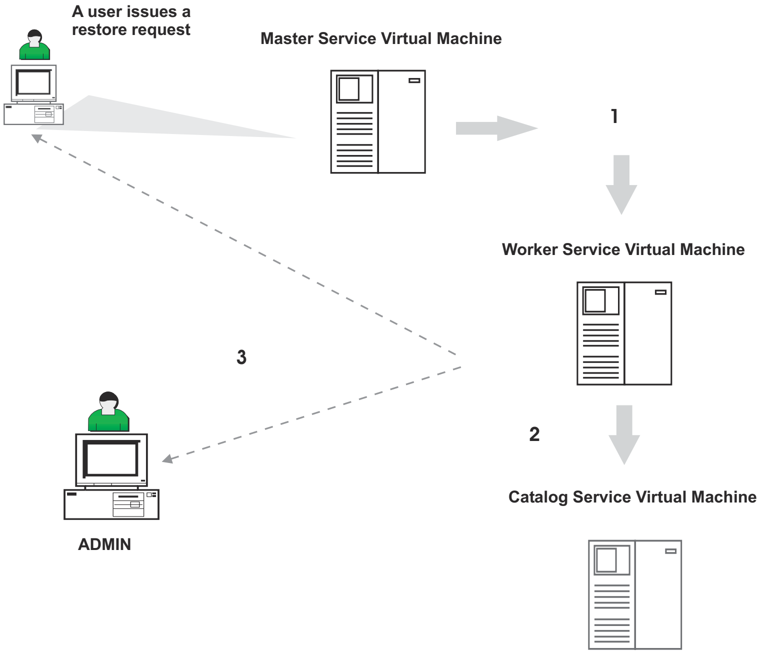

A user issues a restore request
Master Service Virtual Machine
1
Worker Service Virtual Machine
3
2
Catalog Service Virtual Machine
ADMIN

Figure 1. Overview of backup and restore processing

Backup request processing

When a backup administrator issues a backup request, the following actions occur.

1. The primary service virtual machine validates the request. If the request is accepted, a worker service virtual machine is dispatched to process the request.
2. The catalog service virtual machine records the data for inclusion in the backup catalog.
3. The worker service virtual machine informs the administrator of the final result when processing is complete.

For more information about performing backups, see the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*.

Restore request processing

When you issue a restore request, the following actions occur.

1. The primary service virtual machine validates the request. If the request is accepted, a restore job is built and a worker service virtual machine is dispatched to process the restore request.

2. The worker service virtual machine manages the restore operation and informs the ADMIN user and the client of the final result when processing is complete.

Chapter 2. Locating and restoring backup data

Backup and Restore Manager provides a panel interface for locating the data that you previously backed up and several options to restore the data.

Your authority to the backup catalog determines the entries you are able to view. If you have sufficient access, you see all entries. If you have access to your own entries only, you see only your catalog entries. If you do not have access or if the catalog is empty, you cannot view entries and the EXEC stops.

Note: The BKRUSER, BKRVOL, and BKRXNTD catalog browser interface EXECs are designed primarily for administrators. Typically, only administrators have the necessary access to the backup catalog for these EXECs to be useful.

Timestamps reflect the time zone of the local z/VM system.

Complete these steps to locate and restore your data:

1. Locate the data in the backup catalog that you want to restore. Backup and Restore Manager provides several panel interfaces, called catalog browser interfaces, to help you locate the data that was backed up. You invoke a catalog browser interface with one of the following commands:

   - To view catalog contents by job, with the ability to filter by job name, object owner, type of content (SFS, EDF, FBA, or CKD), and object name, use the BKRJOB EXEC. For more information, see “BKRJOB” on page 25.
   - To view catalog contents by file listing with the ability to filter by owner, filename, filetype, or filemode number, use the BKRLIST EXEC. For more information, see “BKRLIST” on page 29.
   - To view catalog contents by user ID, with the ability to drill down to a specific owner ID and see each object that was backed up for the user ID, use the BKRUSER EXEC. For more information, see “BKRUSER” on page 32.
   - To view catalog contents by DASD volume name, with the ability to drill down to a particular volume and view every backed-up object on the volume by: owner, device type, offset from the beginning of the volume, and size, and then additionally filter the view by owner and device address, use the BKRVOL EXEC. For more information, see “BKRVOL” on page 35.
   - To view catalog contents by DASD extent, with the ability to filter by DASD volume name, object owner, device address, and job name, use the BKRXNTD EXEC. For more information, see “BKRXNTD” on page 38.

2. To restore data, ensure that the media type destination is compatible with original media (source) as shown in the following table:

   Table 1. Data compatibility matrix (RESTORE command)

	Destination					
Source	BFS	CKD	FBA	EDF	SFS	RDR
CKD		X				
FBA			X			
EDF				X	X	X
SFS				X	X	X
BFS	X					

   Notes:

   a. X. Compatible (restore permitted).

   b. EDF data can be restored to RDR, EDF, or pre-existing SFS targets.

c. SFS base files can be restored to RDR or EDF targets. Base files, authorizations, and other SFS artifacts such as aliases, external objects, can be restored to any file pool to which the worker service virtual machine has ADMIN privileges.

d. If the SFS target is not already enrolled, it is re-enrolled using the same storage limits as the backed-up file spaces.

e. Restoration of an SFS backup organizes data into the same directory hierarchy as the original file space. Restoration to an alternate file pool or file space is permitted. However, the directory structure that is contained in the originating file space is preserved.

3. Understand how restore processing works for various data types. Restore processing options vary depending on the type of restore you are performing:

   - Restore files or data to an EDF minidisk. For more information, see "Restoring data to an EDF minidisk" on page 7.
   - Restore a DASD volume or minidisk image (CKD or FB-512). For more information, see "Performing a DASD image restore (CKD or FB-512)" on page 8.
   - Restore files to a user reader. For more information, see "Restoring data to a user RDR" on page 9.
   - Restore data to an SFS target. For more information, see "Restoring data to an SFS target" on page 9.
   - Restore a BFS file space. For more information, see "Performing a BFS file space restore" on page 10.

4. Restore data that was backed up. Choose the method that you want to use to restore the data:

   - A catalog browser interface.
   - The **RESTORE** command. For more information, see "RESTORE command" on page 11.
   - Batch restore processing. For more information, see "Batch restore processing" on page 17.
   - Restore application code that you develop using the examples that are provided in Appendix B, "Application code examples," on page 43.

Note: Restore operations that are based on the use of the catalog browser interfaces, the **RESTORE** command, or the batch restore interface rely on backup content that is produced through normal system backup operations as described in the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*. Backup data that is generated by the sample application code is not incorporated into the backup catalog and is not available for restore operations through the catalog browser interface, the **RESTORE** command, or batch restore processing.

Restoring data from a catalog browser interface

Complete these steps to restore data from a catalog browser interface.

Note: The below conditions need to be satisfied for using catalog browser interface commands.

- An A-Disk in R/W mode.
- The target minidisk must not be LINKed R/W by any other user when the RESTORE command is processed.

1. Select from the following catalog browser interfaces:

   - BKRJOB.
   - BKRLIST.
   - BKRUSER.
   - BKRVOL.
   - BKRXNTD.

2. After you locate the data that you want to restore using the catalog browser interface, press PF10 (where available) to restore the selected data.

When you press PF10 on an entry, a corresponding **Restore Specifications** panel is displayed, depending on the data type:

- Files on CMS EDF minidisk
- Files in SFS
- File spaces in BFS
- Image data on CKD or FBA DASD

Notes:

- If the **Restore** option is not available, the following message is displayed:

  ```
  BKR8824E Restore is not available in this view.
  ```

- When you request a restore operation, you must specify the name of the local backup primary user ID. The default user ID is BKRBKUP.

3. From the **Restore Specifications** panel, choose a restore target destination such as EDF, RDR, or SFS, depending on the type of data that you want to restore.

   Note: Only one type of target destination (EDF, RDR, or SFS) is permitted. Specifying multiple types of target destinations is not permitted.

Restoring data to an EDF minidisk

To restore data to an EDF minidisk, complete these steps.

1. In the **To EDF minidisk, userid** field, specify the user ID of the owner of the data that you want to restore (1-8 characters; alphanumeric).
2. In the **and virtual address** field, specify the target restore address (1-4 characters).
3. In the **FORMAT: OK if needed?** field, specify whether the restore operation has permission to format the minidisk that is specified as the restore target, if necessary. Specify one of the following options:

   YES

   The restore operation has permission to format the minidisk that is specified as the restore target, if necessary.

   NO

   The restore operation does *not* have permission to format the minidisk that is specified as the restore target, if necessary.

   Note: The **FORMAT: OK if needed** option applies only if the restore target is an unformatted minidisk.
4. In the **FORMAT regardless?** field, specify whether the restore operation is required to format the minidisk that is specified as the restore target before any files are restored. Specify one of the following options:

   YES

   The restore operation is required to format the minidisk that is specified as the restore target before files are restored.

   NO

   The restore operation is *not* required to format the minidisk that is specified as the restore target before files are restored.

   Note: When you specify YES in the **FORMAT regardless?** field, you must specify YES in the **FORMAT: OK if needed** field. Valid `FORMAT` combinations are described in the following table:

Table 2. Valid FORMAT combinations

FORMAT: OK if needed	**FORMAT regardless**	**Result**
NO	NO	The target minidisk is not formatted. If the minidisk is not already formatted, the restore operation fails.
YES	NO	If the target minidisk was not previously formatted, it can be formatted. If the minidisk is already formatted, the restore operation proceeds. Pre-existing files on the minidisk are retained.
YES	YES	The target minidisk is formatted before the restore. Pre-existing data on the minidisk is destroyed before restore processing.

5. To select a specific file or group of files to restore, specify the following information in the **File filters** fields:

   Filename
   The name of the file to restore (1-8 characters; alphanumeric). Wildcard characters are permitted.

   Filetype
   The file type of the files to restore (1-8 characters; alphanumeric). Wildcard characters are permitted.

   mode number
   The mode number of the files to restore. Valid values are a single numeric (0-6) or a wildcard character (*).

6. In the **Options** field, specify one or more of the following options:

   FROMALT
   Specify `FROMALT` to perform restore operations from an alternate member of an IBMTWIN or DUALTAPE volume pair:

   - If the associated backup job was configured to use the IBMTWIN or DUALTAPE output handler, `FROMALT` forces the restore operation to use the alternate tape volume from a tape pair.
   - The `FROMALT` option has no effect if the original backup was performed using the IBMTAPE or CMSFILE output handlers (it is ignored).

   DELFILES
   (The `DELFILES` option applies only to a restore of backup data that is created during an incremental backup.) When an incremental backup is taken, Backup and Restore Manager creates a record of the files that are deleted from a minidisk or SFS file space after the baseline full backup was performed. If you specify `DELFILES` to restore an incremental backup, recovery processing attempts to re-create the contents of the destination minidisk or file space as it was at the time of the incremental backup by issuing an **ERASE** command for these files.

   REPLACE
   When Backup and Restore Manager encounters a pre-existing file during restore-to-minidisk operations, the REPLACE option causes the pre-existing file to be replaced by the backup version. If you do not specify REPLACE, the default behavior is to skip over pre-existing files.

Performing a DASD image restore (CKD or FB-512)

To perform a DASD image restore, you must have an available minidisk as the target of the restore operation. The minidisk can be the original minidisk from which the backup was taken, or a new minidisk. In both cases, DASD image restores overwrite the data that was previously on the minidisk. To perform a DASD image restore, complete these steps.

1. In the **To userid** field, specify the user ID of the owner of the data that you want to restore (1-8 characters; alphanumeric).
2. In the **and virtual device address** field, specify the target restore address (1-4 characters).
3. In the **Options** field, specify one or more of the following options:

FROMALT

Specify FROMALT to perform restore operations from an alternate member of an IBMTWIN or DUALTAPE volume pair:

- If the associated backup job was configured to use the IBMTWIN or DUALTAPE output handler, FROMALT forces the restore operation to use the alternate tape volume from a tape pair.
- The FROMALT option has no effect if the original backup was performed using the IBMTAPE or CMSFILE output handlers, it is ignored.

Note: DASD image restores overwrite the data that was previously on the device.

Restoring data to a user RDR

To restore data to a user RDR, complete these steps.

1. In the **To userid** field, specify the user ID of the owner of the data to restore (1-8 characters; alphanumeric).
2. In the **node** field, specify the target restore node (1-8 characters). The default is the current node.
3. To select a specific file or group of files to restore, specify the following information in the **File filters** fields:

   Filename

   The name of the file to restore (1-8 characters; alphanumeric). Wildcard characters are permitted.

   Filetype

   The file type of the files to restore (1-8 characters; alphanumeric). Wildcard characters are permitted.

   mode number

   The mode number of the files to restore. Valid values are a single numeric (0-6) or a wildcard character (*).

Restoring data to an SFS target

To restore data to an SFS target, complete these steps.

1. In the **To filepool** field, specify the target SFS file pool (1-8 characters; alphanumeric).
2. In the **filespace** field, specify the target file space (1-8 characters; alphanumeric).
3. In the **path** field, specify the target path.

   Note: This option is available from the **CMS EDF Minidisk Restore Specifications** panel.
4. In the **optional storage group** field, specify an optional storage group (a valid file pool server storage group).

   Note: This option is applicable only if the target file space is not enrolled in the file pool. If the target file space is not enrolled and you omit this option, the original storage group number is used.
5. To select a specific file or group of files to restore, specify the following information in the **File filters** fields:

   Filename

   The name of the file to restore (1-8 characters; alphanumeric). Wildcard characters are permitted.

   Filetype

   The file type of the files to restore (1-8 characters; alphanumeric). Wildcard characters are permitted.

   mode number

   The mode number of the files to restore. Valid values are a single numeric (0-6) or a wildcard character (*).

   Path filter

   The path of the files to restore.
6. In the **Options** field, specify one or more of the following options:

FROMALT
Specify FROMALT to perform restore operations from an alternate member of an IBMTWIN or DUALTAPE volume pair:

- If the associated backup job was configured to use the IBMTWIN or DUALTAPE output handler, FROMALT forces the restore operation to use the alternate tape volume from a tape pair.
- The FROMALT option has no effect if the original backup was performed using the IBMTAPE or CMSFILE output handlers (it is ignored).

DELFILES
(The DELFILES option applies only to a restore of backup data that is created during an incremental backup.) When an incremental backup is taken, Backup and Restore Manager creates a record of the files that are deleted from a minidisk or SFS file space after the baseline full backup was performed. If you specify DELFILES to restore an incremental backup, recovery processing attempts to re-create the contents of the destination minidisk or file space as it was at the time of the incremental backup by issuing an **ERASE** command for these files.

REPLACE
When Backup and Restore Manager encounters a pre-existing file during restore-to-minidisk operations, the REPLACE option causes the pre-existing file to be replaced by the backup version. If you do not specify REPLACE, the default behavior is to skip over pre-existing files.

Note: DASD image restores overwrite the data that was previously on the device.

Performing a BFS file space restore

To perform a BFS file space restore, complete these steps.

1. In the **To filepool** field, specify the target SFS file pool (1-8 characters; alphanumeric).
2. In the **filespace** field, specify the target BFS file space (1-8 characters; alphanumeric).
3. In the **optional storage group** field, specify an optional storage group.

   Note: This option is applicable only if the target file space is not enrolled in the file pool. If the target file space is not enrolled, and you omit this option, the original storage group number is used.

Creating a Restore EXEC

When you use the BKRJOB and BKRXNTD catalog browser interfaces, you can use PF2 to create a Backup and Restore Manager EXEC called BKR EXEC. The format of the EXEC is similar to the CMS EXEC that you can create using LISTFILE for a selected group of objects.

1. Use filters to filter the items that you want to restore.
2. To create an EXEC, press PF2.

   Note: For the BKRJOB and BKRXNTD EXECs, pressing PF2 creates an EXEC.

   Message BKR8814I indicates that the EXEC was created successfully. Figure 2 on page 10 shows an example Backup and Restore Manager EXEC.

```
/* REXX */
Parse Arg Pre '%%' Post;
Address 'CMS';
Pre 'SAMPFULL 00000028 R54TUX01 EDF $DEV0191' Post
Pre 'SAMPFULL 00000036 R54TUX01 EDF $DEV0191' Post
Pre 'SAMPFULL 00000039 R54TUX01 EDF $DEV0191' Post
Pre 'SAMPFULL 00000040 R54TUX01 EDF $DEV0191' Post
Pre 'SAMPFULL 00000041 R54TUX01 EDF $DEV0191' Post
```

Figure 2. Backup and Restore Manager EXEC

3. To facilitate invoking the EXEC with a **RESTORE** command, one line is displayed for each object.
   For example, issue the following command to restore the PROFILE EXEC from every instance to the specified user's reader, or with EXECs of your own creation:

```
"BKR RESTORE %% TO RDR = - PROFILE EXEC *"
```

RESTORE command

The **RESTORE** command restores data from any object type (EDF, SFS, FBA, BFS, or CKD) to any compatible location (RDR, SFS, EDF, FBA, BFS, or CKD).

Note: The Backup and Restore Manager user interface Restore option (PF10) is the preferred method of restoring small amounts of data.

RESTORE command syntax

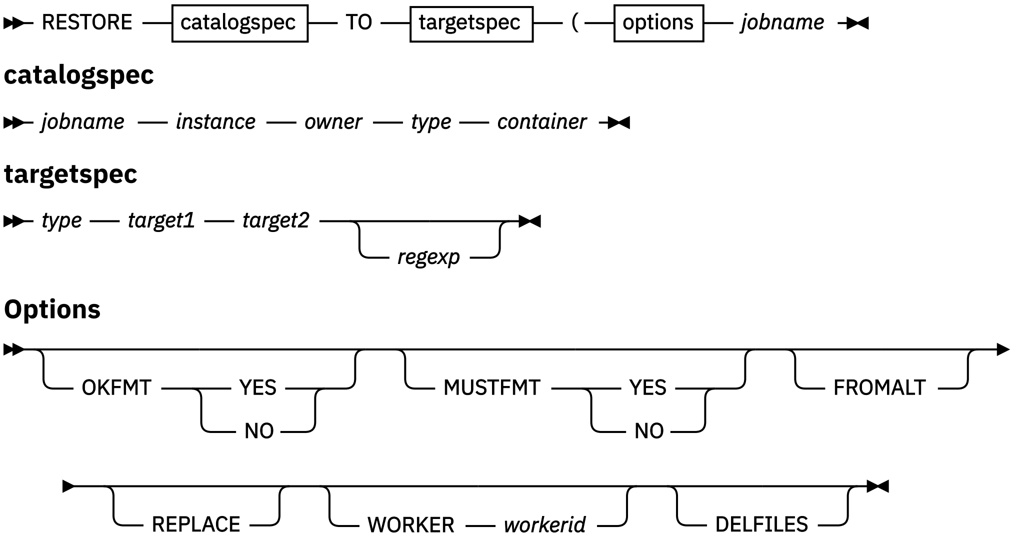


Figure 3. RESTORE command syntax

Authorization

The **RESTORE** command has the following authorization requirements:

- All users can restore data from any object type (EDF, SFS, FBA, or CKD) that is owned by their user ID to any location (RDR, SFS, EDF, FBA, or CKD) that is owned by their user ID.
- Users with ADMIN privileges can restore any backup content to any valid destination.

Note: Special consideration: To restore DASD image backup data to real DASD devices, without first defining a full volume minidisk, administrators must enable an additional target location type (DEV) through a system-wide option. The following considerations apply:

- Use of DEV target locations is controlled in `BKRSYSTM CONFIG` through the **`BKR_Allow_RealDevice_Restore`** parameter. This function is disabled by default.
- Real DASD devices selected as DEV targets for **RESTORE** must be online to CP, but *not* ATTACHed to SYSTEM.

Invoking the command

Invoke the **RESTORE** command in the following ways:

- From the catalog browser interface (the preferred method).
- In a file that is submitted for batch processing. For more information, see "Batch restore processing" on page 17.
- Issued directly using the SMSG interface to BKRBKUP.

Notes:

1. To invoke the **RESTORE** command directly, issue `CP SMSG BKRBKUP RESTORE ...parameters...` (where BKRBKUP is the name of your primary backup service virtual machine).
2. If you are using the catalog browser interface, the **RESTORE** command is not required. The routines invoke the **RESTORE** command for you.

Catalogspec

Specifies the source of the data to restore.

jobname

The name of the backup job that incorporates the data for the restore operation (1-8 characters; alphanumeric).

instance

The specific instance of *jobname* to reference (8 characters, numeric). The valid range is 00000001-99999999.

owner

The VM user ID of the owner of the data to restore (1-8 characters; alphanumeric).

type

The type of data to restore. Specify one of the following values:

BFS

Byte File System.

CKD

CKD track image backup.

EDF

CMS formatted minidisk backup.

FBA

FBA block image backup.

SFS

Shared File System backup.

container

A minidisk (*vdev*) or file space (1 to 8 characters alphanumeric).

Targetspec

The location to which to restore the data.

type

The destination data type. Specify one of the following options:

CKD

Restore raw CKD track image backups to the minidisk that is identified by *userid vdev*. The syntax is:

```
CKD userid vdev
```

Where *vdev* is a valid minidisk address (1-4 hexadecimal digits).

Notes:

1. The target minidisk must be available for WR link at the time the restore request is processed and must not be linked by another user.
2. Because image backup restore handles data at the ECKD track image (or FBA block image) level, the target minidisk does not need to be pre-formatted.

EDF

Restore data to a CMS formatted minidisk that is identified by *userid vdev*. The syntax is:

```
EDF userid vdev
```

Where *vdev* is a valid minidisk address (1-4 hexadecimal digits).

Notes:

1. The target minidisk must be available for WR link at the time the restore request is processed and must not be linked by another user.
2. The target minidisk must be formatted and the EDF block size must be consistent with the source data.

FBA

Restore FBA block-image backup to an FBA device that is identified by *userid vdev*. The syntax is:

```
FBA userid vdev
```

Where *vdev* is a valid minidisk address (1-4 hexadecimal digits).

DEV

Restore a raw CKD or FB-512 image backup to a real DASD device. The syntax is:

```
DEV SYSTEM rdev volser
```

Where:

- *rdev* is a real DASD device (1-4 hexadecimal digits).
- *volser* is the current DASD device volume label or **SCRATCH**.

Note:

- Backup worker service virtual machines (BKRWRK*nn*) must be able to CP ATTACH target devices to themselves.
- The target device must be online and not ATTACHed to SYSTEM.
- The target device must be the same DASD type as the source backup data, and of equal or greater capacity.
- DEV capability is disabled by default. Use of DEV restore targets must be enabled in `BKRSYSTM CONFIG` by specifying:

  ```
  BKR_Allow_RealDevice_Restore = 1
  ```
- The target DASD device current volume label must match the *volser* argument. To use an unlabeled device as a restore target, or to ignore an existing label, specify SCRATCH for the *volser* parameter.
- Take care to avoid overwriting DASD volumes that are shared with other systems or LPARs.
- Take care to avoid duplicating DASD volume labels that are already used by the system. To avoid conflicts, you are responsible for relabeling restored DASD volumes as needed.

RDR

Restore to the reader of *userid*, either at the RSCS node *nodeid* or the local host system that is indicated by a dash (-). The syntax is:

```
RDR userid nodeid|-
```

Note: If *nodeid* is not a dash, the value is used as a remote RSCS-accessible node for a SENDFILE compatible *userid* AT *nodeid* delivery address. A value of "-" causes files to be spooled to *userid* on the local system.

SFS

Restore CMS minidisk or SFS data to SFS.

If you are restoring SFS data to SFS, the syntax is:

```
SFS poolname:spacename sg#
```

This syntax restores SFS data to *poolname:spacename*. If the target file space is not already enrolled on the target file pool, it is re-enrolled with the same limits in effect at the time of backup to the storage group specified by *sg#*.

If you are restoring CMS minidisk data to SFS, the syntax is:

```
SFS poolname:spacename.dir.dir.dir -
```

Where *poolname:spacename.dir.dir.dir* can specify a multi-qualifier directory. For example: `VMSYSU:SYSPROG.RESTORE.MAINT191`.

Note: The dash (-) is required.

Notes:

1. When you restore data that was originally backed up from a CMS formatted minidisk to the specified SFS location, you must create the target directory before you submit the restore request.
2. The target directory must be predefined. The virtual machine that performs the restore operation requires ADMIN privileges for the target pool, or WRITE permission to the target directory.

BFS

Restore a BFS file space image to BFS. If you are restoring a BFS file space, use the following syntax:

```
BFS poolname:spacename sg#
```

target1

The restore target identifier 1 for the *type* operand. See the description of each value for the *type* operand above.

target2

The restore target identifier 2 for the *type* operand. See the description of each value for the *type* operand above.

regexp

A wildcard pattern to filter objects to select for the restore operation:

- If source data is from SFS, the *regexp* syntax consists of *pathmask fnmask ftmask* and *fm#*. Specify '* * * *' to select all objects.
- If source data is from EDF, the *regexp* syntax consists of *fnmask ftmask* and *fm#*. Specify '* * *' to select all objects.
- For CKD and FBA data, omit the *regexp* operand.
- o For BFS, the *regexp* operand must be specified as “*” (a single asterisk).

Note: Wildcard specification for the *regexp* operand is similar to CMS LISTFILE where an asterisk (*) represents one or more characters, and a percent sign (%) represents a single character.

Options

The FORMAT options OKFMT and MUSTFMT apply only to the following restore scenarios.

- A CMS minidisk backup that is restored to a minidisk. If the target mindisk is not formatted, the target minidisk is formatted to match the source minidisk. For example, the same EDF blocksize and CMS minidisk label is restored, in addition to the files that are specified on the **RESTORE** command.
- An SFS file space restored to a minidisk. If the target mindisk is not formatted, the target minidisk is formatted with an EDF block size of 4K (4096 bytes). The minidisk label is set to EDF-4K. (The 4K

blocksize is required to restore SFS data to a CMS minidisk.) The label value is set to EDF-4K because no prior label information can be extracted from the SFS backup information.

OKFMT

Specifies whether the restore operation has permission to FORMAT the minidisk that is specified as the restore target, if necessary. Specify one of the following values:

YES

The restore operation has permission to FORMAT the minidisk that is specified as the restore target, if necessary.

NO

The restore operation does *not* have permission to FORMAT the minidisk that is specified as the restore target, if necessary.

MUSTFMT

Specifies whether the restore operation is required to FORMAT the minidisk that is specified as the restore target before files are restored. Specify one of the following values:

YES

The restore operation is required to FORMAT the minidisk that is specified as the restore target before files are restored.

NO

The restore operation is *not* required to FORMAT the minidisk that is specified as the restore target before files are restored.

Note: When you set MUSTFMT to YES, set OKFMT to YES. Valid combinations are described in Table 3 on page 15.

Table 3. Valid MUSTFMT OKFMT combinations

MUSTFMT	OKFMT	Result
NO	NO	The target minidisk is not formatted. If the minidisk is not formatted, the restore operation fails.
NO	YES	The target minidisk can be formatted if it is not previously formatted. If the minidisk is formatted, the restore operation proceeds. Pre-existing files on the minidisk are retained.
YES	YES	The target minidisk is formatted before restore processing. Pre-existing data on the minidisk is destroyed prior to restore processing.

FROMALT

If the associated backup job was configured to use the IBMTWIN or DUALTAPE output handler, FROMALT forces the restore operation to use the alternate tape volume from a tape pair.

The FROMALT option is ignored if the original backup was performed using the IBMTAPE or CMSFILE output handlers.

REPLACE

When Backup and Restore Manager encounters a pre-existing file during restore to minidisk operations, the REPLACE option causes the pre-existing file to be replaced by the backup version. If you do not specify REPLACE, the default behavior is to skip over pre-existing files.

WORKER *workerid*

Designates a specific worker service virtual machine for processing of the RESTORE function. You can use this option to force multiple **RESTORE** commands to be processed by a single worker. This feature is useful when multiple restore requests are issued for data on the same tape volume.

DELFILES

(This option applies only to the restoration of backup data that is created during incremental backup.) When you take an incremental backup, Backup and Restore Manager creates a record of the files that were deleted from a minidisk or SFS file space after the baseline full backup was performed. If you specify DELFILES to restore from an incremental backup, recovery processing attempts to re-create the contents of the destination minidisk or file space as it existed at the time of the incremental backup by issuing an **ERASE** command for the files.

For example, at the time a full backup is taken, the ABCUSER 191 minidisk contains the following files:

- FILE ONE A
- FILE TWO A
- FILE THREE A

The full backup contains a copy of each of the above files.

Later, when an incremental backup is taken, the state of the minidisk has changed:

- FILE ONE A
- FILE THREE A
- FILE FOUR A

Since the full backup was created:

- FILE ONE A (recently updated)
- FILE TWO A (erased)
- FILE THREE A (unchanged)
- FILE FOUR A (created)

Thus, the incremental backup contains the following items:

- The updated version of FILE ONE A
- A copy of FILE FOUR A
- A record of the deletion of FILE TWO A

To restore the contents of the ABCUSER 191 minidisk to an empty, newly formatted CMS minidisk, the results of the restore operation varies depending on the use of the REPLACE and DELFILES options.

Assume that first, a **RESTORE** command is issued for the full backup. The destination minidisk contains the following files, all from the full backup:

- FILE ONE A
- FILE TWO A
- FILE THREE A

After the full backup is recovered, a restore from the incremental backup is performed. If the **RESTORE** command is issued with no options, the minidisk contains the following files:

- FILE ONE A (the version from the full backup)
- FILE TWO A (the version from the full backup)
- FILE THREE A (the version from the full backup)
- FILE FOUR A (the version from the incremental backup)

If the REPLACE option is specified, the minidisk contains the following files:

- FILE ONE A (the version from the incremental backup replaces the version from the full backup)
- FILE TWO A (the version from the full backup)
- FILE THREE A (the version from the full backup)
- FILE FOUR A (the version from the incremental backup)

If the REPLACE and DELFILES options are specified, the minidisk contains the following files:

- FILE ONE A (the version from the incremental backup replaces the version from the full backup)
- FILE THREE A (the version from the full backup)

- FILE FOUR A (the version from the incremental backup)

In this case, FILE TWO A is not present because the DELFILES option caused restore processing to delete the file.

Batch restore processing

Create batch restore requests through XEDIT. The **RESTORE** command syntax that is used with batch restore processing is identical to the **RESTORE** command syntax that is used with the catalog browser interface.

For batch restore operations, specify options using one of the following methods:

- **RESTORE** command options are delimited by a left parenthesis "(" as part of the first **RESTORE** command in a batch set.
- Alternately, specify an OPTION statement as the first record in a set of batch **RESTORE** commands.

Note the following items:

- **RESTORE** options are in effect for the entire batch operation. Defined options are in effect for all **RESTORE** commands in a batch, whether specified with an OPTION statement or as part of the first **RESTORE** command in a set.
- The statements in a set of batch mode **RESTORE** commands are subject to multiple line continuation with syntax that is similar to REXX. To continue a line, specify a trailing comma "," as the last character in a record.
- Blank lines are permitted between **RESTORE** commands in the batch restore job.
- Comments are not permitted in a batch restore job.
- You must send the file that contains batch mode **RESTORE** commands to BKRBKUP in SENDFILE (netdata) format. The file name may be any valid alphanumeric CMS file name (1-8 characters, A-Z, and 0-9). The CMS file type must be JOB. The batch restore job must be transmitted as CP spool class J. For example:

  ```
  SENDFILE RESTORE JOB A TO BKRBKUP (CLASS J
  ```

- Access control for batch **RESTORE** mode restore operations is based on the spool file origin ID. The user-of-origin for the spool file that contains batch mode **RESTORE** commands must be authorized for each **RESTORE** command in the batch file. If the user that issues a restore request is not authorized for one or more of the commands in the request, the specific commands are rejected and the rest of the commands are processed. For more information, see "RESTORE command" on page 11.
- The worker service virtual machine considers a batch restore request to be one job. Mounted tape is retained after each **RESTORE** command is processed and is unloaded only if the next **RESTORE** command in the batch request requires a different tape. The **Tape_Retain_After_EOJ** configuration option is only in effect at the end of the batch request. For more information, see the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*.

The following examples show a batch **RESTORE** command. The examples are submitted for processing using **SENDFILE** from a user that is granted backup ADMIN privileges in the BKRUSERS NAMES file:

```
SENDFILE RESTORE JOB A TO BKRBKUP (CLASS J
```

Example 1: Restore a single backup

The following example shows how to restore a single backup. It shows use of the OPTION statement and continuation handling.

```
 RESTORE  JOB      A1  V 80  Trunc=80 Size=24 Line=0 Col=1 Alt=0
Editing existing file...






===== * * * Top of File * * *
      T...T....T....T....T....T....T....T....T....T....T....T....T....T....T...
===== OPTION ,
=====  WORKER BKRWRK01 ,
=====  REPLACE
=====
===== RESTORE SAMPFULL 00000170 ,
=====         OPMGRM1 EDF $DEV0194 ,
=====         TO EDF SYSPROG 9405 ,
=====         * * *
=====
===== * * * End of File * * *
```

Figure 4. Example 1 (Restore a single backup)

The restore job is processed by worker BKRWRK01. The job consists of one **RESTORE** command that restores all CMS minidisk files to the SYSPROG 9405 minidisk from backup job SAMPFULL, instance 00000170, that were backed up from the user ID OPMGRM1 194 disk.

Example 2: Restoring multiple backups

The following example shows how to restore multiple backups.

```
 RESTORE  JOB      A1  V 80  Trunc=80 Size=24 Line=0 Col=1 Alt=0
Editing existing file...






===== * * * Top of File * * *
      T...T....T....T....T....T....T....T....T....T....T....T....T....T....T...
===== OPTION ,
=====  WORKER BKRWRK01 ,
=====  REPLACE
=====
===== RESTORE SAMPFULL 00000170 ,
=====         OPMGRM1 EDF $DEV0194 ,
=====         TO EDF SYSPROG 9405 ,
=====         * * *
=====
===== RESTORE SAMPINCR 00000659 ,
=====         OPMGRM1 EDF $DEV0194 ,
=====         TO EDF SYSPROG 9405 ,
=====         * * *
=====
===== RESTORE SAMPFULL 00000170 ,
=====         SYSPROG SFS VMSYSU ,
=====         TO SFS VMSYSU:SYSPROGA 2 ,
=====         * * * *
=====
===== RESTORE SAMPINCR 00000659 ,
=====         SYSPROG SFS VMSYSU ,
=====         TO SFS VMSYSU:SYSPROGA 2 ,
=====         * * * *
=====
===== * * * End of File * * *
```

Figure 5. Example 2 (Restoring multiple backups)

In the example, BKRWRK01 processes all of the restore requests. Restore processing replaces any file that exists in the target location with the file from the backup.

BKRBRT command (Bulk Restore Tool)

The BKRBRT command generates batch-mode job statements to restore data from the most recent backup jobs, based on specified criteria, to a previous location. This data is either owned by an individual user or is located on MDISKS from a specific real DASD volume. For CMS file-level backups, RESTORE commands may be generated for both a baseline FULL file-level backup and a related INCREMENTAL file backup.

The restore job statements generated by BKRBRT are not automatically submitted for processing. The statements are generated and written to the console and/or a file on disk. You can then modify the statements as needed before submitting them using the batch restore interface. Refer to "Batch restore processing" on page 17 for more information on submitting the job statements.

BKRBRT command syntax

```
▶▶─ BKRBRT ── source ── ( ── options ─▶◀
```

Figure 6. BKRBRT command

Authorization

System Administrator

Operands

Any of these operands can be used in this command.

source

Identifies the source of data to be restored. The default is the current z/VM user ID. Valid values are:

<blank>

If no value is specified, *source* defaults to the current user ID, which is the user invoking BKRBRT.

user_id

If a non-blank value is specified, and the "VOLSER" option is not used, *source* will be interpreted as a z/VM user ID, an SFS file space name, or a BFS file space name.

Note: Normally a CMS user will only own a single 191 minidisk. However, If a user owns multiple minidisks, the RESTORE commands generated by BKRBRT can be edited to remove any minidisk(s) or SFS file spaces that you want to exclude from being restored. You may also modify the "TO" portion of the RESTORE command to specify a different destination.

volume_label

If a non-blank value is specified and the "VOLSER" option is used, *source* will be interpreted as a real DASD volume label. This combination of options will cause BKRBRT to search for backups of all MDISKs defined on the specified real DASD volume label.

options

Many optional parameters are available to use with this command. Each option is described below.

VOLSER

The *source* is treated as a real DASD volume label. BKRBRT will search the backup catalog for all MDISKs defined on that volume. RESTORE commands will be generated for each MDISK defined on that volume, subject to any filtering imposed by other options. If a VOLSER is not specified, the default is the z/VM user ID.

Note: If the VOLSER option is not specified, *source* is treated as a z/VM user ID, an SFS file space name, or a BFS file space name.

NOSFS
Exclude SFS (Shared File System) file spaces from RESTORE processing.

Note: The default is SFS.

NOBFS
Exclude BFS (Byte File System) file spaces from RESTORE processing.

Note: The default is BFS.

NOCKD
Exclude ECKD DASD image backups from RESTORE processing.

Note: The default is CKD.

NOFBA
Exclude FBA/FB-512 DASD image backups from RESTORE processing.

Note: The default is FBA.

NOIMAGE
Exclude both ECKD and FBA/FB-512 DASD image backups from RESTORE processing. NOIMAGE has the same affect as specifying both NOCKD and NOFBA.

Note: The default is IMAGE.

NOEDF
Exclude CMS file-level backups of CMS EDF minidisk file systems from RESTORE processing.

Note: The default is EDF.

JOBNAME *job_spec*
Restrict the set of backup jobs searched during catalog inspection to those produced by backup jobs named *job_spec*. For example, to confine the search to backups created by job SAMPFULL, specify `JOBNAME SAMPFULL`. The default is an asterisk (*), which indicates that any job name can be used.

The *job_spec* can be a LISTFILE-compatible wild card expression. For example, specifying `"JOBNAME SAMP*"` would include backups created by jobs named `SAMPFULL` and `SAMPINCR`.

BEFORE *time_spec*
Restrict the set of backup jobs searched during catalog inspection to those created on or **before** the *time spec*, which contains two parts: *before_date before_time*.

If BEFORE is not specified, or if the *before_date* parameter is omitted, the default behavior is to NOT apply any "BEFORE" filtering. If the *before_date* is a valid date (yyyy/mm/dd), and *before_time* is not specified, *before_time* defaults to 00:00:00 (i.e. midnight on *before_date*).

AFTER *time_spec*
Restrict the set of backup jobs searched during catalog inspection to those created on or **after** the *time spec*, which contains two parts: *after_date after_time*.

If AFTER is not specified, or if the *after_date* parameter is omitted, the default behavior is to NOT apply any "AFTER" filtering. If the *after_date* is a valid date (yyyy/mm/dd), and *after_time* is not specified, *after_time* defaults to 23:59:59 (the ending hour:minute:second of *after_date*).

CONSOLE/TYPE
The default is that BKRBRT will display output on the console.

NOCONSOLE
BKRBRT console output will be suppressed. The default is BKRBRT will display output on the console.

TYPE
Use as a synonym for CONSOLE.

NOTYPE
Use as a synonym for NOCONSOLE.

time_spec

A one- or two-token parameter that declares a specific date, or date and time of day. Format is as follows:

yyyy/mm/dd

This format uses a four-digit year, /, a two-digit month, /, and a two-digit day. For example, to specify December 31st of 2020, specify `2020/12/31`.

yyyy/mm/dd hh:mn:ss

This format uses a four-digit year, /, a two-digit month, /, a two-digit day, which is `yyyy/mm/dd`, then one or more spaces followed by the 24-hour format hour, :, minutes, :, and seconds, which is `hh:mm:ss`. For example, to specify June 1, 2020, at 12:00:00, specify `2020/06/01 12:00:00`.

Note: Additional information for *time_spec*:

- A two-digit year, such as "20" for "2020", can be specified. Two-digit years are assumed to be 21st-century dates. For example, `20/06/01` will be treated as `2020/06/01`.
- Values for "mm" (month) and "dd" (day) can be single-digit integers, such as 2020/6/1. Values will be padded with zeros to create the equivalent two-digit values.

FILE

If specified, RESTORE statements will be written to CMS file $BKRBRT `JOB A`. If this file already exists, BKRBRT will exit with a warning message (BKRBRT9619W) and return code 4.

EDIT

If specified, RESTORE statements will be written to CMS file $BKRBRT `JOB A`. If this file already exists, *RESTORE commands will be* ***appended*** *to $BKRBRT JOB A*.

When BKRBRT completes, XEDIT will automatically be invoked to edit $BKRBRT `FILE A`. This behavior allows a Backup Administrator to build and customize a single batch restore job using the results of multiple BKRBRT searches.

Example 1: Use BKRBRT to restore all minidisks for user TS9341 to the most recent contents in the backup catalog.

In this example, it is assumed that BKRBRT is invoked to create batch RESTORE statements to restore the most recent full and incremental backups of user TS9341. Also user TS9341 only has a 191 minidisk and there are no SFS or BFS filespaces owned by this user. The most recent full backup of TS9431 MDISK 191 was created on June 28, 2020. After that full backup was created, an incremental backup was created on June 29, 2020.

To restore the most recent version of files for TS9341 191, two batch-mode RESTORE statements must be executed. Because BKRBRT was invoked with the default options, all results are displayed on the console. See the example below.

Figure 7. BKRBRT with RESTORE statements

```
bkrbrt ts9431
BKRBRT9614I BKRBRT invoked with default parameters.
BKRBRT9616I Querying backup catalog for containers owned by user TS9431
BKRBRT9617I Catalog query discovered backups of 1 unique data containers.

* Unfiltered catalog query results included data from minidisk TS9431 0191
* Source data: Full file-level EDF backup created 06/28/2020 04:18:17 of ts9431 0191
RESTORE SAMPFULL 00000711 ,
        TS9431 EDF 0191 ,
        TO EDF TS9431 0191 * * *
* Source data: Incremental file-level EDF backup created 06/29/2020 02:16:28 of TS9431 0191
RESTORE SAMPINCR 00003824 ,
        TS9431 EDF 0191 ,
Ready;
```

Example 2: Use BKRBRT with FILE and NOTYPE to send RESTORE statements to a CMS file.

This example is similar to "Example 1" above, except that BKRBRT is invoked with options "FILE" and "NOTYPE". The resulting batch RESTORE statements are written to CMS file $BKRBRT JOB A, but are not displayed on the console.

At this point, a Backup Administrator has the option of editing the $BKRBRT JOB to add an OPTION statement at the start of the job, or to modify the restore destination ("TO EDF TS9431 0191 * * *") portion of the RESTORE commands to direct restore results to a different destination. The Backup Administrator can submit the batch RESTORE commands for processing by executing the CMS command ("SENDFILE $BKRBRT JOB A TO BKRBKUP (CLASS J").

Figure 8. BKRBRT with FILE and NOTYPE options

```
bkrbrt ts9431 (file notype
BKRBRT9615I BKRBRT invoked with user parameters "FILE NOTYPE"
BKRBRT9616I Querying backup catalog for containers owned by user TS9431
BKRBRT9617I Catalog query discovered backups of 1 unique data containers.
Ready;

listfile $bkrbrt job a (label
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS   DATE     TIME  LABEL
$BKRBRT  JOB      A1 V         93         10          1  7/31/20 14:26:24 DM292Z
Ready;


TYPE $BKRBRT JOB A

* Unfiltered catalog query results included data from minidisk TS9431 0191
* Source data: Full file-level EDF backup created 06/28/2020 04:18:17 of TS9431 0191
RESTORE SAMPFULL 00000711 ,
        TS9431 EDF 0191 ,
        TO EDF TS9431 0191 * * *
* Source data: Incremental file-level EDF backup created 06/29/2020 02:16:28 of TS9431 0191
RESTORE SAMPINCR 00003824 ,
        TS9431 EDF 0191 ,
        TO EDF TS9431 0191 * * *

Ready;
```

Appendix A. Catalog browser interface routines

To use a catalog browser interface, provide the EXEC that you want to use with input (such as the catalog path to search, or, if you are requesting a restore operation, the name of the local backup primary user ID).

Note: The below conditions need to be satisfied for using catalog browser interface commands.

- An A-Disk in R/W mode.
- The target minidisk must not be LINKed R/W by any other user when the RESTORE command is processed.

Invoking a catalog browser interface

To use a catalog browser interface, provide the EXEC that you want to use with input (such as the catalog path to search, or, if you are requesting a restore operation, the name of the local backup master/primary user ID).

- Provide input to a catalog browser interface EXEC in the following ways:
  - Ensure the configuration file, BKRSYSTM CONFIG, is available on any of your accessed minidisks or SFS directories. The EXECs locate the configuration file, parse it, and then extract the required information. To invoke the EXEC when you use this method, type the name of the EXEC you want to invoke (for example: BKRVOL) and press Enter.

    Note: You do not require read-write access to the BKRSYSTM CONFIG file. Read-only access is sufficient.
  - Specify an alternately named backup configuration file name as a command line parameter. For example:

    ```
    BKRVOL (MYBKUP CONFIG *
    ```
  - Specify the catalog path and local backup master (or primary) user ID on the command line. For example:

    ```
    BKRVOL ( MYCAT:MYSPACE. MYBKUPID
    ```

Supported wildcard characters for filtering

To specify the data you want to view, catalog browser interface EXECs support wildcard characters where filtering is permitted. The following table lists the supported wildcard characters and shows examples of wildcard filtering.

Table 4. Catalog browser interface supported wildcard characters

Wildcard character	Description	Example
Asterisk (*)	Represents zero or more of any character.	To view entries that have owner IDs that begin with "SM" (such as SM123 or SMUSER2), specify SM* for an *Ownerid* filter.
Percent sign (%)	Represents one character.	To view entries with owner IDs that begin with "SM" and end with one character (such as SM1 or SMM), specify SM% for an *Ownerid* filter.
Number sign (#)	Represents one numeric character (0-9).	To view entries with owner IDs that begin with "SM" and end with one numeric character (such as SM1 or SM5), specify SM# for an *Ownerid* filter.

Table 4. Catalog browser interface supported wildcard characters (continued)

Wildcard character	Description	Example
At sign (@)	Represents one hexadecimal character (A-F, a-f, 0-9).	To view entries with owner IDs that begin with "SM" and end with one hexadecimal character (such as SMA or SM3), specify `SM@` for an *Ownerid* filter.
Ampersand (&)	Represents one alphabetic (A-Z, a-z) character.	To view entries with owner IDs that begin with "SM" and end with one alphabetic character (such as SMA or SMD), specify `SM&` for an *Ownerid* filter.
Double quotation mark (")	The double quotation mark is an escape character that is used to treat one of the other special characters literally.	To view entries with owner IDs that begin with "SM" and end with a numeric character (such as SM123 or SMMID5), specify `SM*#` for an *Ownerid* filter.

Note:

- If no data passes the filters, a message is displayed.
- Because all backup catalog data is uppercase, all filters are translated to uppercase before they are applied.

How backup instances are treated

Because backup instances are numeric, they are treated differently for a command-line filter than job names or owner names. If there is no wildcard character in the filter, then the value is padded on the left with zeros, if necessary, to eight characters and matches only the resulting specific instance number. If a wildcard character is found, then the instance is treated like any other filter string ("3" is padded to "00000003" and matches only that instance number). "01*3" is treated as a regular expression and matches an instance number that begins with "01" and ends with "3".

Impact of command line filters on performance

The command line allows filtering by job name, instance name, or owner. After the initial display, you can further filter the results by job name, owner, type, or device.

Command line filters can help or impact performance, depending on the SFS authority of the person that invokes them and how they are used. Users with SFS administrative authority see the most impact.

In general, every non-wildcard filter that is specified on the command line, from left to right, improves performance because it allows the catalog browser interface EXEC to restrict its access to the backup catalog to a smaller view. Wildcard filters do not allow the catalog view to be restricted, and incur extra processing. For example:

"BKRJOB ABCDEFGH" performs faster than:

"BKRJOB", which performs faster than:

"BKRJOB AB*GH"

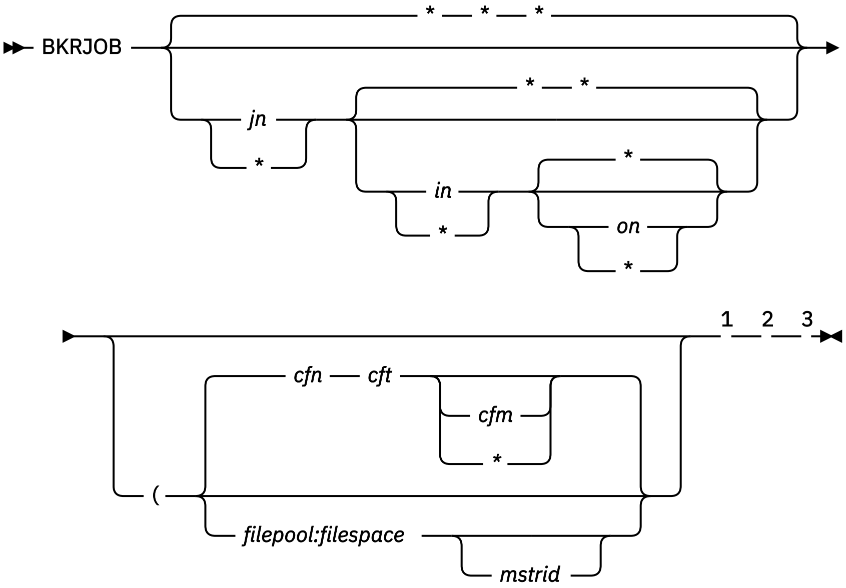
PF key definitions

Where data is presented in vertical columns with one horizontal line per instance, sort the data in ascending or descending order by placing the cursor in the field on which you want to sort and then pressing PF5 or PF6.

PF7 scrolls backward and PF8 scrolls forward. PF3 is Quit. PF11 drills down to the next available level of detail until no more levels are available. For more information, see Appendix A, “Catalog browser interface routines,” on page 23.

BKRJOB

The **BKRJOB** command displays information such as backup jobs, instances, owners, and devices in the backup job catalog.



Notes:

[1] If you invoke BKRJOB without configuration parameters (*cfn*, *cft*, *cfm*), it searches for the default configuration file `BKRSYSTM CONFIG` on any accessed filemode.

[2] If you invoke BKRJOB with an alternate configuration file parameter (*cfn*, *cft*), *cfm* defaults to * if not specified.

[3] If you invoke BKRJOB with *filepool:filespace*, you cannot issue commands such as **RESTORE** unless the appropriate backup primary server user ID (*mstrid*) is provided.

Figure 9. BKRJOB syntax

Authorization

BKRJOB displays the portion of the Backup and Restore Manager catalog that you are authorized to view.

Operands

jn

The view of the backup catalog is restricted to the job names that match the job names that are specified on this parameter. Wildcard characters are supported.

in

The view of the backup catalog is restricted to instances that match the instances that are specified on this parameter. Wildcard characters are supported.

on

The view of the backup catalog is restricted to owners that match the owners that are specified on this parameter. Wildcard characters are supported.

cfn

The file name of an alternative configuration file.

cft

The file type of an alternative configuration file.

cfm

The file mode of an alternative configuration file. If you do not specify a mode, the default is * (any accessed file mode).

filepool

The SFS catalog file pool to browse for backup jobs, instead of obtaining the root from a backup configuration file.

Note: You must specify *filepool* with *filespace*.

filespace

The SFS file space to browse for backup jobs, instead of obtaining the file pool from a backup configuration file.

Note: You must specify *filespace* with *filepool*.

mstrid

(Optional.) The backup primary server user ID to which to issue commands when an SFS root and file pool are provided, instead of obtaining the information from a backup configuration file. No **RESTORE** commands can be issued when BKRJOB is invoked with *filepool:filespace* unless you specify *mstrid*.

Usage notes

1. Initially, no information is displayed in the Date and Time columns because obtaining it might be performance intensive if you select a large number of granules. To display the date and time of completion for the displayed lines, press PF9.

   If the operation takes more than 20 seconds of elapsed time, you are prompted to continue or end the operation. If you choose to continue, you are prompted after each additional 20 seconds of elapsed time. If you choose to end the operation, the information that was obtained is displayed.

   Date and time values are not changed or refreshed if the filters are altered. If, after obtaining date and time information for one subset of granules, you subsequently change your filter settings, you might see some lines with date and time values and some without.

   Date and time are considered to be a single field for sorting purposes. If you sort on date and time, ensure that you obtain the date and time for all of the lines that are currently displayed (PF9).

2. To perform a common operation against a large group of granules, create an exec (PF2).

   BKR EXEC is created on the disk or directory that is accessed as A, with the following first three lines:

   ```
        /* REXX */
        Parse Arg Pre '%%' Post;
        Address 'CMS'
   ```

   These lines are followed by one line of the following form for each line that is currently displayed in the file:

   ```
   Pre 'jobname instance owner type device' Post;
   ```

   The format allows you to invoke a command for each granule with arbitrary strings preceding or following the granule information, provided the strings do not contain two consecutive percent signs.

   For example, the following command issues a **RESTORE** command for all exec files from each granule, and specifies them to be sent to the reader of USER1 on the issuing node:

   ```
   BKR RESTORE %% TO RDR USER1 - * EXEC *
   ```

3. Lines are initially displayed as they are returned from the catalog inquiry. Use the sort keys (PF5 and PF6) or filters (job name, instance, owner, type, or device) to change the lines that are displayed and the order in which they are displayed.

4. To restrict the display to subsets of the items you are authorized to view, use filters. You can use filters on the command line when you invoke BKRJOB, or you can specify filters on the panel after the initial display. See "Supported wildcard characters for filtering" on page 23 for more information.

   Note: All filters default to * (no filtering occurs).

5. Issue commands directly from the line on which a catalog granule is displayed. Commands are passed to CMS and normal CMS command resolution is used. The forward slash (/) controls the substitution of information from the displayed line into the command as shown in the following table:

   Table 5. Symbol substitution

Symbol	Description
/	The job name, instance, owner, type, and device is displayed on the line.
/j	The job name is displayed on the line.
/i	The instance is displayed on the line.
/o	The owner is displayed on the line.
/t	The type is displayed on the line.
/d	The device is displayed on the line.

   Specify the symbols in any combination, or order, and repeat them as needed. For example:

   - `/j /o` indicates that the job name is followed by the object. Because the symbols are not immediately adjacent, a space is added in front of the object.
   - `/i/d` indicates that the instance is followed by the device (no intervening spaces).

   If you do not specify forward slash symbols, Backup and Restore Manager appends the job name, instance, owner, type, and device to the command, with one intervening space in front of each.
6. If you specify a set of filters which excludes all lines, the message `No lines passed filters` and all lines in the file are displayed.
7. Because all backup catalog data is uppercase, Backup and Restore Manager translates all filters to upper case before they are applied.

PF key definitions

Table 6 on page 27 describes the BKRJOB PF key definitions.

Table 6. BKRJOB PF key definitions

Key	Setting	Action
Enter	Run	Run the command (or commands) type on file lines.
PF1	Help	Display the **BKRJOB** command description.
PF2	Create exec	Create the file BKR EXEC on the disk or directory that is accessed as A, The file contains one line with the contents of each object that meets the current filters. To perform a common operation against a selected set of granules, you can use this REXX EXEC. For more information, see "Usage notes" on page 26.
PF3	Quit	Exit from BKRJOB.
PF4	Return	Return to the previous level, or exit if at the highest level.
PF5	Sort up	Sort the displayed lines in ascending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF6	Sort down	Sort the displayed lines in descending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF7	Backward	Scroll back one panel.
PF8	Forward	Scroll forward one panel.

Table 6. BKRJOB PF key definitions (continued)

Key	Setting	Action
PF9	Show times	Extract and display the date and time of completion (the SFS last-change date and time) for each granule that passes the current filters. **Note:** If there are many granules that are selected, this command might take some time to complete.
PF11	Display	If the cursor is on a line that displays a granule, detailed information from that granule is extracted and displayed.

Messages

Table 7 on page 28 lists the BKRJOB messages.

Table 7. BKRJOB messages

Message number	Message
BKR8807E	No catalog name was supplied.
BKR8808E	No entries in the catalog, or the catalog is not accessible.
BKR8809E	Unexpected catalog error *rc*; the list might be incomplete.
BKR8810E	Error *rc* closing catalog.
BKR8811E	No entries were returned for volume search.
BKR8812E	No records passed filters.
BKR8813E	Exec creation failed, EXECIO RC=*rc*.
BKR8814E	BKR EXEC successfully written.
BKR8815E	The cursor is not within a valid sort field.
BKR8816E	The cursor is not on a file line.
BKR8817W	Over (*n*) seconds elapsed for this operation.
BKR8818R	Do you want to continue? (Yes|No).
BKR8819I	Terminating date/time retrieval.
BKR8820E	No granule name was passed to expand.
BKR8821E	Error expanding granule file.
BKR8822E	*(value)* is not a valid line command.
BKR8825E	The cursor is not on a valid selection.

Return codes

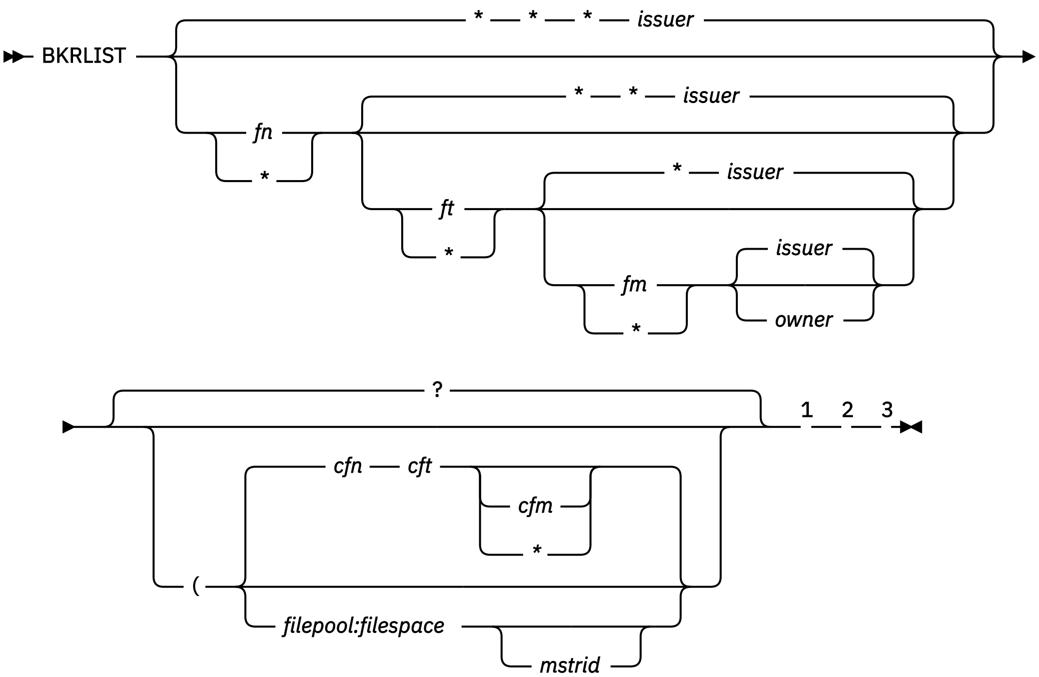
Table 8 on page 28 shows return codes for BKRJOB.

Table 8. BKRJOB return codes

Return code	Description
0	Finished correctly.
4	No entries in the catalog, or the catalog is not accessible.
12	No catalog name was supplied.

BKRLIST

BKRLIST displays the portion of the backup catalog you are authorized to view. The **BKRLIST** command displays information about backup jobs, instances, owners, and devices within the backup job catalog.



Notes:

[1] If you invoke BKRLIST without parameters, it searches for the default configuration file BKRSYSTM CONFIG on any accessed filemode.

[2] If you invoke BKRLIST with an alternate configuration file parameter (*cfn*, *cft*), *cfm* defaults to an asterisk (*) if not specified.

[3] If you invoke BKRLIST with *filepool:filespace*, you cannot issue commands such as **RESTORE** unless the appropriate backup primary server user ID (*mstrid*) is also provided.

Figure 10. BKRLIST syntax

Operands

fn
The file names to display. Wildcard characters are permitted.

ft
The file types to display. Wildcard characters are permitted.

fm
The file modes to display. Specify an asterisk (*) or a number (0-6).

owner
The file owners to display. Wildcard characters are permitted.

issuer
The user ID from which BKRLIST is run.

cfn
The file name of an alternate configuration file.

cft
The file type of an alternate configuration file.

cfm

The file mode of an alternate configuration file. If you do not specify a mode, the default is * (any accessed filemode).

filepool

An SFS catalog file pool to browse for backup jobs (rather than obtaining the file pool from a backup configuration file).

Note: You must specify *filepool* with *filespace*.

filespace

The SFS file space to browse for backup jobs, rather than obtaining the file space from a backup configuration file.

Note: You must specify *filespace* with *filepool*.

mstrid

(Optional.) The backup primary server user ID to which commands are issued when *filepool* and *filespace* are provided (as opposed to obtaining the information from a backup configuration file). No **RESTORE** commands can be issued when BKRLIST is invoked with *filepool:filespace* unless you specify *mstrid*.

Usage notes

1. BKRLIST displays the portion of the backup job catalog that you are authorized to see, subject to the filtering of *fn*, *ft*, *fm*, or *owner* operands on the command line.
2. The default (no operands are specified on the command line) limits the view to the items that are owned by the user ID that issued the **BKRLIST** command. This behavior improves performance through a smaller catalog search.
3. If your user ID authority allows you to view all, or a large part of the backup catalog, and you specify an *owner* operand using wildcard characters, it can require significant real time, processing time, and virtual storage to complete the search. The search can produce a large result file. Therefore, it is suggested that users such as administrators avoid the use of wildcards in *owner* specifications, where possible.
4. Backup and Restore Manager displays lines as they are returned from the catalog inquiry. To change the lines that are displayed, use the filters (file name, file type, file mode, or owner). To change the order in which the lines display, use the sort options (PF5 and PF6).
5. Filter the initial selection by using the *fn*, *ft*, *fm*, and *owner* on the command line. You can further filter your results within BKRLIST by using the name, type, mode, and owner filters on the initial BKRLIST panel. (All of the fields on the initial BKRLIST panel default to *, therefore the initial display consists of all of the lines that meet the command line filters.) You can filter on *filename*, *filetype*, *filemode*, or *owner*. In all cases, the filter is a regular expression in which various arbitrary characters can be used. For more information, see "Supported wildcard characters for filtering" on page 23.
6. If you specify a set of filters that excludes all lines, the message `No lines passed filters` and all lines in the file are displayed.
7. Because all backup catalog data is uppercase, Backup and Restore Manager translates all filters to uppercase before they are applied.

PF key definitions

Table 9 on page 30 describes the BKRLIST PF key definitions.

Table 9. BKRLIST PF key definitions

Key	Setting	Action
Enter	Run	Run the command (or commands) type on file lines.
PF1	Help	Display the **BKRLIST** command description.

Table 9. BKRLIST PF key definitions (continued)

Key	Setting	Action
PF2	Create exec	Create the file BKR EXEC on the disk or directory that is accessed as A. The file contains one line with the contents of each object that meets the current filters. To perform a common operation against a selected set of granules, use this REXX EXEC. See "Usage notes" on page 30 for a description of the BKR EXEC file.
PF	Quit	Exit from BKRLIST.
PF4	Return	Return to the previous level, or exit if at the highest level.
PF5	Sort up	Sort the displayed lines in ascending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF6	Sort down	Sort the displayed lines in descending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF7	Backward	Scroll back one panel.
PF8	Forward	Scroll forward one panel.
PF9	Show times	Extract and display the date and time of completion (the SFS last-change date and time) for each granule that passes the current filters. **Note:** If many granules are selected, this command might take some time to complete.
PF10	Restore	If the cursor is on a line that displays an instance, the **RESTORE** panel is invoked for the instance.
PF11	Display	If the cursor is on a line that displays a granule, detailed information from that granule is extracted and displayed.

Messages

Table 10 on page 31 lists the BKRLIST messages.

Table 10. BKRLIST messages

Message number	Message
BKR8807E	No catalog name was supplied.
BKR8808E	No entries in the catalog, or the catalog is not accessible.
BKR8809E	Unexpected catalog error *rc*. The list might be incomplete.
BKR8810E	Error *rc* closing catalog.
BKR8811E	No entries were returned for the volume search.
BKR8812E	No records passed filters.
BKR8815E	The cursor is not within a valid sort field.
BKR8816E	The cursor is not on a file line.
BKR8820E	No granule name was passed to expand.
BKR8821E	Error expanding granule file.
BKR8827E	File too large. Out of storage.

Return codes

Table 11 on page 32 shows the BKRLIST return codes.

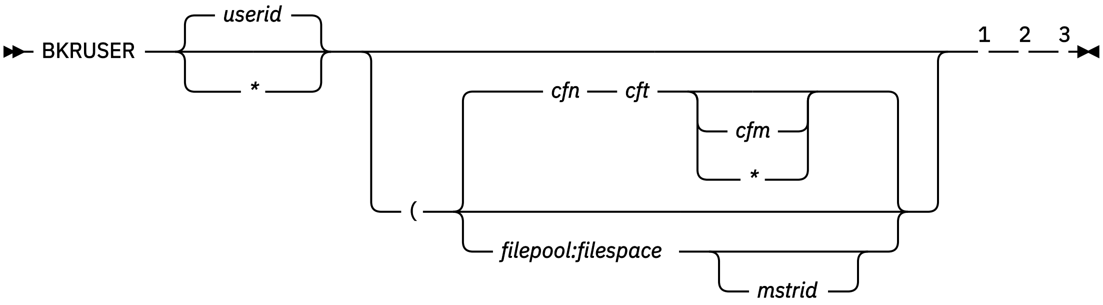
Table 11. BKRLIST return codes

Return code	Description
0	Finished correctly.
4	No entries in the catalog, or the catalog is not accessible.
12	No catalog name was supplied.

BKRUSER

BKRUSER displays users for whom Backup and Restore Manager has catalog data. The list of users is restricted to those user IDs the issuing user is authorized to see within the scope of the **`userid`** command line parameter. The **`userid`** value defaults to the issuing user ID. You can override the default by specifying another user ID, or by specifying * (all users).

Note: This command is intended primarily for administrators. Typically, only administrators have the required access to the backup SFS catalog.



Notes:

[1] If you invoke BKRUSER without configuration file parameters (*cfn*, *cft*, *cfm*), BKRUSER searches for default configuration file `BKRSYSTM CONFIG` on any accessed filemode.

[2] If you invoke BKRUSER with an alternate configuration file parameter (*cfn*, *cft*), *cfm* defaults to * if not specified.

[3] If you invoke BKRUSER with *filepool:filespace*, no commands such as **RESTORE** can be issued unless the appropriate backup primary server user ID (*mstrid*) is provided.

Figure 11. BKRUSER syntax

Operands

userid

Restrict the view of the backup catalog to instances of data that have been backed up and are owned by the specified user ID. Wildcard characters are not supported.

The operand defaults to the issuing user ID. Specify an asterisk * to view instances that belong to all users.

cfn

The file name of an alternative configuration file.

cft

The file type of an alternative configuration file.

cfm

The file mode of an alternative configuration file. If you do not specify a mode, the default is * (any accessed filemode).

filepool

The SFS catalog file pool to browse for backup jobs, instead of obtaining the file pool from a backup configuration file.

Note: You must specify *filepool* with *filespace*.

filespace

The SFS file space to browse for backup jobs, instead of obtaining the file space from a backup configuration file.

Note: You must specify *filespace* with *filepool*.

mstrid

(Optional.) The backup primary server user ID to which to issue commands when *filepool* and *filespace* are provided (instead of obtaining the information from a backup configuration file). No **RESTORE** commands can be issued when BKRUSER is invoked with *filepool:filespace* unless you specify *mstrid*.

Usage notes

1. Specifying an asterisk for the *userid* operand might require significant system resources and cause increased response time if the Backup and Restore Manager catalog is large and the invoking user has SFS administrative authority to the catalog. IBM recommends specifying a user ID, or defaulting to the invoking user ID, whenever possible.
2. Lines are initially displayed as they are returned from the catalog inquiry. You can use sorting (PF 5 and PF 6) or the filters (jobname, owner, type, or device) to change both the lines are displayed and the order in which they are displayed.
3. You can restrict the display to various subsets of what the initial catalog inquiry returns by using filters. Initially all filters are set to "*", which matches all lines.

   From the **Ownerids** panel, you can filter by owner ID. From the **Devices** panel, you can filter by device and type. On the lowest level panel, which displays the jobs that have backed up a particular object, you can filter on job name. In all cases, the filter is a regular expression in which various arbitrary characters can be used as described in "Supported wildcard characters for filtering" on page 23.

   If you specify a set of filters that excludes all lines, the message `No lines passed filters` and all lines in the file are displayed.
4. Because all backup catalog data is uppercase, Backup and Restore Manager translates all filters to uppercase before they are applied.

PF key definitions

Table 12 on page 33 describes the BKRUSER PF key definitions.

Table 12. BKRUSER PF key definitions

Key	Setting	Action
Enter	Run	Run the command (or commands) type on file lines.
PF1	Help	Display the **BKRUSER** command description.
PF2	Create EXEC	Create the file BKR EXEC on the disk or directory that is accessed as A. The file contains one line with the contents of each object that meets the current filters. To perform a common operation against a selected set of granules, use this REXX EXEC. See "Usage notes" on page 33 for a description of the BKR EXEC file.
PF	Quit	Exit from BKRUSER.
PF4	Return	Return to the previous level, or exit if at the highest level.

Table 12. BKRUSER PF key definitions (continued)

Key	Setting	Action
PF5	Sort up	Sort the displayed lines in ascending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF6	Sort down	Sort the displayed lines in descending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF7	Backward	Scroll back one panel.
PF8	Forward	Scroll forward one panel.
PF9	Show times	Extract and display the date and time of completion (the SFS last-change date and time) for each granule that passes the current filters. **Note:** If you select many granules, this command might take some time to complete.
PF11	Display	If the cursor is on a line which shows a granule, detailed information from that granule is extracted and displayed.

Messages

Table 13 on page 34 lists the BKRUSER messages.

Table 13. BKRUSER messages

Message number	Message
BKR8807E	No catalog name was supplied.
BKR8808E	No entries in the catalog, or the catalog is not accessible.
BKR8809E	Unexpected catalog error *rc*; the list might be incomplete.
BKR8810E	Error *rc* closing catalog.
BKR8811E	No entries returned for the volume search.
BKR8812E	No records passed filters.
BKR8813E	EXEC creation failed, EXECIO RC=*rc*.
BKR8814E	BKR EXEC successfully written.
BKR8815E	The cursor is not within a valid sort field.
BKR8816E	The cursor is not on a file line.
BKR8817W	Over (*number*) seconds have elapsed for this operation.
BKR8818R	Do you wish to continue? (Yes|No)
BKR8819I	Terminating date/time retrieval.
BKR8820E	No granule name was passed to expand.
BKR8821E	Error expanding granule file.
BKR8822E	*(value)*is not a valid line command.
BKR8825E	Cursor is not on a valid selection.

Return codes

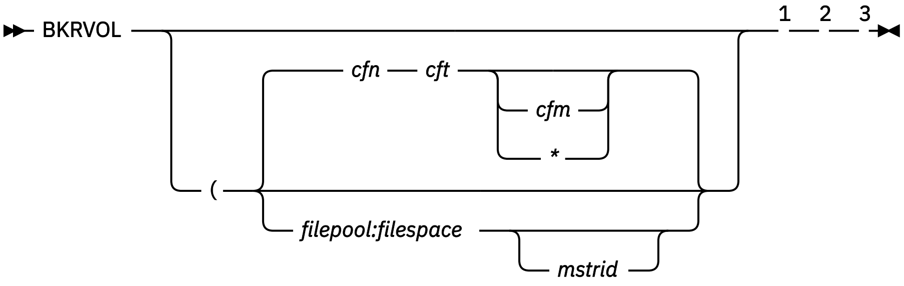
Table 14 on page 35 shows the BKRUSER return codes.

Table 14. BKRUSER return codes

Return code	Description
0	Finished correctly.
4	No entries in the catalog ,or the catalog is not accessible.
12	No catalog name was supplied.

BKRVOL

BKRVOL displays the portion of the backup catalog you are authorized to view. BKRVOL displays all instances of data that is backed up from one or more specific volumes.



Notes:

[1] If you invoke BKRVOL without parameters, it searches for the default configuration file BKRSYSTM `CONFIG` on any accessed filemode.

[2] If you invoke BKRVOL with an alternate configuration file parameter (*cfn*, *cft*), *cfm* defaults to * if not specified.

[3] If you invoke BKRVOL with *filepool:filespace*, you cannot issue commands such as **RESTORE** unless the appropriate backup primary server ID (*mstrid*) is provided.

Figure 12. BKRVOL syntax

Note: This command is intended primarily for administrators. Typically, only administrators have the required access to the backup SFS catalog.

Operands

cfn
The file name of an alternative configuration file.

cft
The file type of an alternative configuration file.

cfm
The file mode of an alternative configuration file. If you do not specify a mode, the default is * (any accessed filemode).

filepool
The SFS catalog file pool to browse for backup jobs, instead of obtaining the root from a backup configuration file.

Note: You must specify *filepool* with *filespace*.

filespace
The SFS file space to browse for backup jobs, instead of obtaining the file space from a backup configuration file.

Note: You must specify *filespace* with *filepool*.

mstrid

(Optional.) The backup primary server user ID to which to issue commands when a file space and file pool are provided, instead of obtaining the information from a backup configuration file. No **RESTORE** commands can be issued when BKRVOL is invoked with *filepool:filespace* unless you specify *mstrid*.

Usage notes

1. If your installation disabled extent catalog creation (through the **`BKR_Catalog_ExtentCat_Enabled`** option in the `BKRSYSTM CONFIG` file), BKRVOL does not display information.
2. Once a volume is selected, Backup and Restore Manager displays lines as they are returned from the catalog inquiry. To change the lines that are displayed and the order in which they display, use the sort options (PF5 and PF6) or the filters (owner or device).
3. Use filters to restrict the display to various subsets of the initial catalog inquiry. Initially all filters are set to "*", which matches all lines.

   You can filter on volume, owner, or device. The filter is a regular expression in which various arbitrary characters can be used as described in "Supported wildcard characters for filtering" on page 23.

PF key definitions

Table 15 on page 36 describes the BKRVOL PF key definitions.

Table 15. BKRVOL PF key definitions

Key	Setting	Action
Enter	Run	Run the command (or commands) type on file lines.
PF1	Help	Display the **BKRVOL** command description.
PF2	Create EXEC	Create the file BKR EXEC on the disk or directory that is accessed as A The file contains one line with the contents of each object that meets the current filters. To perform a common operation against a selected set of granules, you can use the REXX EXEC. A description of the BKR EXEC file is provided in "Usage notes" on page 36.
PF	Quit	Exit from BKRVOL.
PF4	Return	Return to the previous level, or exit if at the highest level.
PF5	Sort up	Sort the displayed lines in ascending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF6	Sort down	Sort the displayed lines in descending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF7	Backward	Scroll back one panel.
PF8	Forward	Scroll forward one panel.
PF9	Show times	Extract and display the date and time of completion (the SFS last-change date and time) for each granule that passes the current filters. **Note:** If you select many granules, this command might take some time to complete.
PF11	Display	If the cursor is on a line that displays a granule, detailed information from that granule is extracted and displayed.

Messages

Table 16 on page 37 lists the BKRVOL messages.

Table 16. BKRVOL messages

Message number	Message
BKR8807E	No catalog name was supplied.
BKR8808E	No entries in the catalog, or the catalog is not accessible.
BKR8809E	Unexpected catalog error *rc*. The list might be incomplete.
BKR8810E	Error *rc* closing catalog.
BKR8811E	No entries returned for the volume search.
BKR8812E	No records passed filters.
BKR8813E	Exec creation failed, EXECIO RC=*rc*.
BKR8814E	BKR EXEC successfully written.
BKR8815E	The cursor is not within a valid sort field.
BKR8816E	The cursor is not on a file line.
BKR8817W	Over (*n*) seconds have elapsed for this operation.
BKR8818R	Do you want to continue? (Yes|No).
BKR8819I	Terminating date/time retrieval.
BKR8820E	No granule name passed to expand.
BKR8821E	Error expanding granule file.
BKR8822E	*(value)* is not a valid line command.
BKR8825E	The cursor is not on a valid selection.

Return codes

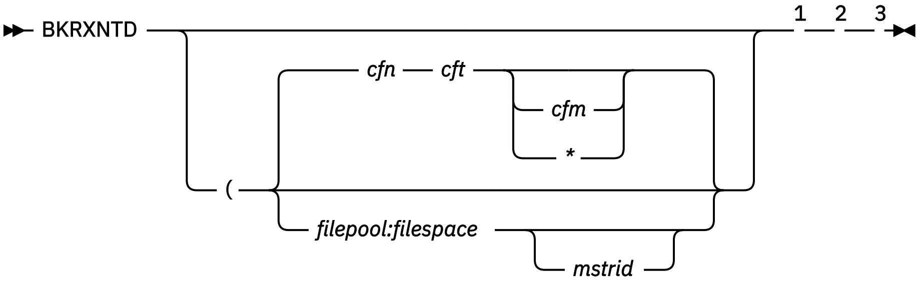
Table 17 on page 37 shows the BKRVOL return codes.

Table 17. BKRVOL return codes

Return code	Description
0	Finished correctly.
4	No entries in the catalog, or the catalog is not accessible.
12	No catalog name was supplied.

BKRXNTD

BKRXNTD displays the portion of the backup catalog you are authorized to view. BKRXNTD displays all instances of backed-up EDF or CKD data arranged by position on DASD volumes.



Notes:

[1] If you invoke BKRXNTD without parameters, it searches for the default configuration file BKRSYSTM CONFIG on any accessed filemode.

[2] If you invoke BKRXNTD with an alternate configuration file parameter (*cfn*, *cft*), *cfm* defaults to * if not specified.

[3] If you invoke BKRXNTD with *filepool:filespace*, you cannot issue commands such as **RESTORE** unless the appropriate backup primary server ID (*mstrid*) is provided.

Figure 13. BKRXNTD syntax

Note: This command is intended primarily for administrators. Typically, only administrators have the required access to the backup SFS catalog.

Operands

cfn
The file name of an alternative configuration file.

cft
The file type of an alternative configuration file.

cfm
The file mode of an alternative configuration file. If you do not specify a mode, the default is * (any accessed filemode).

filepool
The SFS catalog file pool to browse for backup jobs, instead of obtaining the file pool from a backup configuration file.

Note: You must specify *filepool* with *filespace*.

filespace
The SFS file space to browse for backup jobs, instead of obtaining the file space from a backup configuration file.

Note: You must specify *filespace* with *filepool*.

mstrid
(Optional.) The backup primary server user ID to which to issue commands when a file space and file pool are provided, instead of obtaining the information from a backup configuration file. No **RESTORE** commands can be issued when BKRXNTD is invoked with *filepool:filespace* unless *mstrid* is specified.

Usage notes

1. If your installation disabled extent catalog creation (through the **BKR_Catalog_ExtentCat_Enabled** option in the BKRSYSTM CONFIG file), BKRXNTD does not display information.

2. To perform a common operation against a large group of granules, create an exec (PF2).

   BKR EXEC is created on the disk or directory that is accessed as A, with the following first three lines:

   ```
         /* REXX */
         Parse Arg Pre '%%' Post;
         Address 'CMS'
   ```

   These lines are followed by one line of the following form for each line that is currently displayed in the file:

   ```
   Pre 'jobname instance owner type device' Post;
   ```

   The format allows you to invoke a command for each granule with arbitrary strings preceding or following the granule information, provided the strings do not contain two consecutive percent signs.

   For example, the following command issues a **RESTORE** command for all exec files from each granule, and specifies them to be sent to the reader of USER1 on the issuing node:

   ```
   BKR RESTORE %% TO RDR USER1 - * EXEC *
   ```

3. Backup and Restore Manager displays lines as they are returned from the catalog inquiry. To change the lines that are displayed and the order in which they display, use the sort options (PF5 and PF6) or the filters (volume, owner, device, or job name).
4. Use filters to restrict the display to various subsets of the initial catalog inquiry. Initially all filters are set to "*", which matches all lines.

   You can filter on volume, owner, device, or job name. The filter is a regular expression in which various arbitrary characters can be used as described in "Supported wildcard characters for filtering" on page 23.

5. Issue commands directly from the line on which a catalog granule is displayed. Commands are passed to CMS and normal CMS command resolution is used. The forward slash (/) controls the substitution of information from the displayed line into the command as shown in the following table:

   Table 18. Symbol substitution

Symbol	Description
/	The job name, instance, owner, type, and device is displayed on the line.
/j	The job name is displayed on the line.
/i	The instance is displayed on the line.
/o	The owner is displayed on the line.
/t	The type is displayed on the line.
/d	The device is displayed on the line.

   Specify the symbols in any combination, or order, and repeat them as needed. For example:

   - `/j /o` indicates that the job name is followed by the object. Because the symbols are not immediately adjacent, a space is added in front of the object.
   - `/i/d` indicates that the instance is followed by the device (no intervening spaces).

   If you do not specify forward slash symbols, Backup and Restore Manager appends the job name, instance, owner, type, and device to the command, with one intervening space in front of each.

PF key definitions

Table 19 on page 40 describes the BKRXNTD PF key definitions.

Table 19. BKRXNTD PF key definitions

Key	Setting	Action
Enter	Run	Run the command (or commands) type on file lines.
PF1	Help	Display the **BKRXNTD** command description.
PF2	Create EXEC	Create the file BKR EXEC on the disk or directory that is accessed as A, The file contains one line with the contents of each object that meets the current filters. To perform a common operation against a selected set of granules, you can use this REXX EXEC. A description of the BKR EXEC file is provided in "Usage notes" on page 38.
PF	Quit	Exit from BKRXNTD.
PF4	Return	Return to the previous level, or exit if already at the highest level.
PF5	Sort up	Sort the displayed lines in ascending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF6	Sort down	Sort the displayed lines in descending order by the column (job, instance, owner, type, device, or time or date) on which the cursor is located.
PF7	Backward	Scroll back one panel.
PF8	Forward	Scroll forward one panel.
PF9	Show times	Extract and display the date and time of completion (the SFS last-change date and time) for each granule which passes the current filters. **Note:** If there are a large number of granules selected, this command might take some time to complete.
PF11	Display	If the cursor is on a line which shows a granule, detailed information from that granule is extracted and displayed.

Messages

Table 20 on page 40 lists the BKRXNTD messages.

Table 20. BKRXNTD messages

Message number	Message
BKR8807E	No catalog name was supplied.
BKR8808E	No entries in the catalog, or the catalog is not accessible.
BKR8809E	Unexpected catalog error *rc*. The list might be incomplete.
BKR8810E	Error *rc* closing catalog.
BKR8811E	No entries returned for the volume search.
BKR8812E	No records passed filters.
BKR8813E	Exec creation failed, EXECIO RC=*rc*.
BKR8814E	BKR EXEC successfully written.
BKR8815E	The cursor is not within a valid sort field.
BKR8816E	The cursor is not on a file line.

Table 20. BKRXNTD messages (continued)

Message number	**Message**
BKR8817W	Over (*n*) seconds elapsed for this operation.
BKR8818R	Do you want to continue? (Yes|No).
BKR8819I	Terminating date/time retrieval.
BKR8820E	No granule name was passed to expand.
BKR8821E	Error expanding granule file.
BKR8822E	*(value)* is not a valid line command.
BKR8825E	The cursor is not on a valid selection.

Return codes

Table 21 on page 41 shows the BKRXNTD return codes.

Table 21. BKRXNTD return codes

Return code	**Description**
0	Finished correctly.
4	No entries in the catalog, or the catalog is not accessible.
12	No catalog name supplied.

Appendix B. Application code examples

This information provides descriptions of application code that you can develop and use to call backup and restore routines.

For information about Backup and Restore Manager input/output handlers (output methods), see Appendix D, “Input/Output handlers,” on page 75.

Backup application code examples

This information describes the application code that can be used to call backup routines to backup a single CKD or FBA minidisk extent, CMS formatted minidisk, or single SFS file space to tape or to a CMS file.

CKDDUMP syntax

Back up a single CKD minidisk extent to a tape or to a CMS file.

```
►►─ CKDDUMP ── ownerid ── vdev ── ( ── METHOD ── method_name ── method_parms ── TOKEN ─►

   ►── token ── JOBSEQ ── jobseq ─►◄
```

Figure 14. CKDDUMP syntax

Operands

ownerid
The virtual machine ID of the minidisk owner (1-8 characters; alphanumeric).

vdev
The virtual address of the target minidisk.

Options

METHOD *method_name*
The output method. Specify one of the following values:

CMSFILE
For input or output to a CMS file on minidisk or SFS.

IBMTAPE
For input or output to any tape drive supported by CMS.

IBMTWIN
For output that is directed to tape "twin sets."

method_parms
(See “Method parameters” on page 44 for descriptions.)

TOKEN *token*
The job token value (1 - 8 character string).

JOBSEQ *jobseq*
Specifies how to manage the job when you batch multiple backup jobs. Specify one of the following options:

ONLY
Mount the tape at initialization and unmount it at termination. Specifying ONLY obtains a tape mount at initialization, and a tape unmount at termination (the task is expected to manage all tape handling within the single dump instance).

Note: Specify ONLY for a single invocation of a single dump task.

FIRST

Mount the tape at initialization and do not unmount it at termination. Specifying FIRST causes the tape to be mounted at initialization and omits tape unmount on termination (the tape is mounted and left in position upon dump completion).

INTERMED

Do not mount or unmount the tape. Specifying INTERMED assumes that tape is already mounted and positioned by the previous dump task, and this task is not expected to unmount the current tape (that is, the tape is in position; do not reposition the tape or unmount the tape).

LAST

Do not mount the tape at initialization. Unmount the tape at termination. Specifying LAST causes the tape to be closed and unmounted at termination (that is, tape is in position; emit a double tape-mark and unmount the tape).

Note: FIRST, INTERMED, and LAST apply only if the job consists of two or more dump tasks.

Method parameters

IBMTAPE

Direct the output to tape.

volser

The tape volume label (or SCRATCH) of the tape to mount.

file#

The file number.

RW/RO

The access to tape: RO (read-only) or RW (read/write).

IBMTWIN

Direct the output to tape "twin sets." IBMTWIN produces logically identical pairs of tape volumes instead of delivering output to a single tape volume. IBMTWIN produces duplicate media pairs for backup in a single backup run, allowing you to retain one set of tapes onsite and a second set of tapes offsite.

privol

SCRATCH or the specific VOL1 label to mount for output (primary twin set member).

offset

The FSF offset from the VOL1 label for each member of the twin set.

rwstat

For output. This parameter must be set to RW (generically, **rwstat** can be RO or RW).

secvol

SCRATCH or the specific VOL1 label to mount for output (secondary twin set member).

CMSFILE

Direct the output to a CMS file.

filename

The file name of a CMS file to use for output (1-8 characters; alphanumeric).

filetype

The file type of a CMS file to use for output (1-8 characters; alphanumeric).

filemode

The file mode of a CMS file to use for output.

EDFDUMP syntax

Back up a single CMS formatted minidisk to a tape or to a CMS file.

```
►►─ EDFDUMP ── ownerid ── vdev ── fnmask ── ftmask ── fm#mask ── ( ── METHOD ─►

   ►── method_name ── method_parms ── TOKEN ── token ── JOBSEQ ── jobseq ─►◄
```

Figure 15. EDFDUMP syntax

Operands

ownerid
The virtual machine ID of the minidisk owner (1-8 characters; alphanumeric).

vdev
The virtual address of the target minidisk.

fnmask
The wildcard selection mask for the name of the file. To specify a subset of file names, use wildcard characters.

ftmask
The wildcard selection mask for the type of file. To specify a subset of file types, use wildcard characters.

fm#mask
The file mode. Specify an asterisk (*) to indicate any file mode (default) or specify a file mode number.

For information about wildcard characters, see the *IBM z/VM CMS Commands and Utility Reference (SC24-6166)*.

Options

METHOD *method_name*
The output method: `IBMTAPE` (for input or output to any tape drive supported by CMS) or `CMSFILE` (for input or output to a CMS file on minidisk or SFS).

method_parms
(See "Method parameters" on page 46 for descriptions.)

TOKEN *token_value*
The job token value (1 - 8 character string).

JOBSEQ *jobseq*
Specifies how to manage the job when you batch multiple backup jobs. Specify one of the following options:

ONLY
Mount the tape at initialization and unmount it at termination. Specifying `ONLY` obtains a tape mount at initialization, and a tape unmount at termination (the task is expected to manage all tape handling within the single dump instance).

Note: Specify `ONLY` for a single invocation of a single dump task.

FIRST
Mount the tape at initialization and do not unmount it at termination. Specifying `FIRST` causes the tape to be mounted at initialization and omits tape unmount on termination (the tape is mounted and left in position upon dump completion).

INTERMED
Do not mount or unmount the tape. Specifying `INTERMED` assumes that tape is mounted and positioned by the previous dump task, and this task is not expected to unmount the current tape (that is, the tape is in position; do not reposition the tape or unmount the tape).

LAST

Do not mount the tape at initialization. Unmount the tape at termination. Specifying LAST causes the tape to be closed and unmounted at termination (that is, tape is in position; emit a double tape-mark and unmount the tape).

Note: `FIRST`, `INTERMED`, and `LAST` apply only if the job consists of two or more dump tasks.

Method parameters

IBMTAPE

Direct the output to tape.

volser

The tape volume label (or SCRATCH) of the tape to mount.

file#

The file number.

RW/RO

The access to tape. Specify RO (read-only) or RW (read-write).

IBMTWIN

Direct the output to tape "twin sets." `IBMTWIN` produces logically identical pairs of tape volumes instead of delivering output to a single tape volume. `IBMTWIN` produces duplicate media pairs for backup in a single backup run, allowing you to retain one set of tapes on-site and a second set of tapes offsite.

privol

SCRATCH or the specific VOL1 label to be mounted for output (primary twin set member).

offset

The FSF offset from the VOL1 label for each member of the twin set.

rwstat

For output. This parameter must be set to RW (generically, **`rwstat`** can be RO or RW).

secvol

SCRATCH or the specific VOL1 label to be mounted for output (secondary twin set member).

CMSFILE

Direct the output to a CMS file.

filename

The file name of a CMS file that is used for output (1-8 characters; alphanumeric).

filetype

The file type of a CMS file that is used for output (1-8 characters; alphanumeric).

filemode

The file mode of a CMS file that is used for output.

FBADUMP syntax

Back up a single FBA minidisk extent to a tape or to a CMS file.

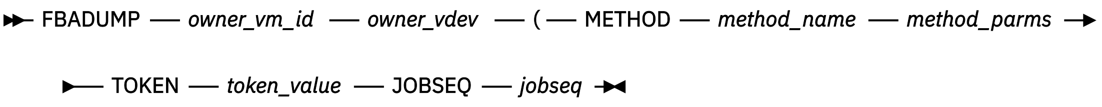


Figure 16. FBADUMP syntax

Operands

owner_vm_id

The virtual machine ID of the minidisk owner (1-8 characters; alphanumeric).

owner_vdev
The virtual address of the target minidisk.

Options

METHOD *method_name*
The output method. Specify one of the following values:

CMSFILE
For input or output to a CMS file on minidisk or SFS.

IBMTAPE
For input or output to any tape drive supported by CMS.

IBMTWIN
For output directed to tape "twin sets."

method_parms
(See "Method parameters" on page 47 for descriptions.)

TOKEN *token_value*
The job token value (1 - 8 character string).

JOBSEQ *jobseq*
Specifies how to manage the job when you batch multiple backup jobs. Specify one of the following values:

ONLY
Mount the tape at initialization and unmount it at termination. Specifying `ONLY` obtains a tape mount at initialization, and a tape unmount at termination (the task is expected to manage all tape handling within the single dump instance).

Note: Specify `ONLY` for a single invocation of a single dump task.

FIRST
Mount the tape at initialization and do not unmount it at termination. Specifying `FIRST` causes the tape to be mounted at initialization and omits tape dismount on termination (the tape is mounted and left in position upon dump completion).

INTERMED
Do not mount or unmount the tape. Specifying `INTERMED` assumes tape is mounted and positioned by the previous dump task, and this task is not expected to unmount the current tape (the tape is in position; do not reposition the tape or unmount the tape).

LAST
Do not mount the tape at initialization. Unmount the tape at termination. Specifying LAST causes the tape to be closed and unmounted at termination (the tape is in position; emit a double tape-mark and unmount the tape).

Note: `FIRST`, `INTERMED`, and `LAST` apply only if the job consists of two or more dump tasks.

Method parameters

IBMTAPE
Direct the output to tape.

volser
The tape volume label (or SCRATCH) of the tape to be mounted.

file#
The file number.

RW/RO
The access to tape: RO (read-only) or RW (read-write).

IBMTWIN
Direct the output to tape "twin sets." Specifying IBMTWIN produces logically identical pairs of tape volumes instead of delivering output to a single tape volume. IBMTWIN produces duplicate media

pairs for backup in a single backup run, allowing you to retain one set of tapes on-site and a second set of tapes offsite.

privol
SCRATCH or the specific VOL1 label to be mounted for output (primary twin set member).

offset
The FSF offset from the VOL1 label for each member of the twin set.

rwstat
For output. This parameter must be set to RW (generically, it can be RO or RW).

secvol
SCRATCH or the specific VOL1 label to be mounted for output (secondary twin set member).

CMSFILE
Direct the output to a CMS file.

filename
The file name of a CMS file to use for output (1-8 characters; alphanumeric).

filetype
The file type of a CMS file to use for output (1-8 characters; alphanumeric).

filemode
The file mode of a CMS file to use for output.

SFSDUMP syntax

Back up a single SFS file space to a tape or to a CMS file.

```
►► SFSDUMP ── poolname ── spacename ── pathmask ── fnmask ── ftmask ── fm#mask ── ( ─►

  ►── METHOD ── method_name ── method_parms ── TOKEN ── token ── JOBSEQ ── jobseq ─►◄
```

Figure 17. SFSDUMP syntax

Operands

poolname
The SFS file pool name.

spacename
The file space name or owner VM name (1-8 characters; alphanumeric).

pathmask
The directory path selection mask.

fnmask
The wildcard selection mask for the name of the file. To specify a subset of file names, use wildcard characters.

ftmask
The wildcard selection mask for the type of file. To specify a subset of file types, use wildcard characters.

fm#num
The file mode. Specify an asterisk (*) to indicate any file mode (default) or specify a file mode number.

For information about wildcard characters, see the *IBM z/VM CMS Commands and Utility Reference (SC24-6166)*.

Options

METHOD *method_name*
The output method. Specify `IBMTAPE` (for input or output to any tape drive supported by CMS) or `CMSFILE` (for input or output to a CMS file on minidisk or SFS).

method_parms

See “Method parameters” on page 49 for descriptions.

TOKEN *token_value*

The job token value (1 - 8 character string).

JOBSEQ *jobseq*

Specifies how to manage the job when you batch multiple backup jobs. Specify one of the following values:

ONLY

Mount the tape at initialization and unmount it at termination. Specifying `ONLY` obtains a tape mount at initialization, and a tape unmount at termination (the task is expected to manage all tape handling within the single dump instance).

Note: Specify `ONLY` for a single invocation of a single dump task.

FIRST

Mount the tape at initialization and do not unmount it at termination. Specifying `FIRST` causes the tape to be mounted at initialization and omits tape dismount on termination (the tape is mounted and left in position upon dump completion).

INTERMED

Do not mount or unmount the tape. Specifying `INTERMED` assumes tape is already mounted and positioned by the previous dump task, and this task is not expected to unmount the current tape (the tape is in position; do not reposition the tape or unmount the tape).

LAST

Do not mount the tape at initialization. Unmount the tape at termination. Specifying LAST causes the tape to be closed and unmounted at termination (the tape is in position; emit a double tape-mark and unmount the tape).

Note: `FIRST`, `INTERMED`, and `LAST` apply only if the job consists of two or more dump tasks.

Method parameters

IBMTAPE

Direct the output to tape.

volser

The tape volume label (or SCRATCH) of the tape to be mounted.

file#

The file number.

RW/RO

The access to tape. RO (read-only) or RW (read/write).

IBMTWIN

Direct output to tape "twin sets." Specifying IBMTWIN produces logically identical pairs of tape volumes instead of delivering output to a single tape volume. IBMTWIN produces duplicate media pairs for backup in a single backup run, allowing you to retain one set of tapes on-site and a second set of tapes offsite.

privol

SCRATCH or the specific VOL1 label to be mounted for output (primary twin set member).

offset

The FSF offset from the VOL1 label for each member of the twin set.

rwstat

For output. This parameter must be set to RW (generically, **`rwstat`** can be RO or RW).

secvol

SCRATCH or the specific VOL1 label to be mounted for output (secondary twin set member).

CMSFILE

Direct the output to a CMS file.

filename
The file name of a CMS file to use for output (1-8 characters; alphanumeric).

filetype
The file type of a CMS file to use for output (1-8 characters; alphanumeric).

filemode
The file mode of a CMS file to use for output.

Restore application code examples

This information provides descriptions of application code (CKDLOAD, EDFLOAD, FBALOAD, and SFSLOAD) that can be used to call restore routines.

CKDLOAD syntax

Restore a raw CKD image backup to disk.

```
►► CKDLOAD — vdev — ( — METHOD — method_name — method_parms ►◄
```

Figure 18. CKDLOAD syntax

Operands

vdev
Specifies the virtual address of the target extent.

Options

METHOD *method_name*
Specifies the method (IBMTAPE or CMSFILE).

method_parms
See "Method parameters" on page 50 for descriptions.

Method parameters

IBMTAPE
Indicates that a tape should be used for the input stream.

volser
Specifies the tape volume label (or SCRATCH) of the tape to be mounted.

file#
Specifies the tape file to be read as input (the label is file #0; the first stream is file #1 and so on).

RW/RO
Specifies the access: RO (read-only) or RW (read-write).

CMSFILE
Indicates that a CMS file should be used for the input stream. The CMSFILE method can be used for input streams sourced from a currently accessed CMS formatted minidisk or SFS directory.

filename
Specifies the file name of a CMS file used for input (1-8 characters; alphanumeric).

filetype
Specifies the file type of a CMS file used for input (1-8 characters; alphanumeric).

filemode
Specifies the file mode of a CMS file used for input.

DDLLOAD syntax

Restore files that were backed up to the spool.

```
►► DDLLOAD — destuser — destnode — pamask — fnmask — ftmask — fmmask — ( ►

   ►— METHOD — method_name — method_parms ►◄
```

Figure 19. DDLLOAD syntax

Operands

destuser
Specifies the destination user for the restore.

destnode
Specifies the destination RSCS/NJE node name.

pamask
Specifies the wildcard selection mask for the SFS data by directory path.

fnmask
The wildcard selection mask for the name of the file.

ftmask
The wildcard selection mask for the type of file.

fmmask
Specifies file mode. Specify an asterisk (*) to indicate any file mode (default) or specify a file mode number.

See the *IBM z/VM CMS Commands and Utility Reference (SC24-6166)* for information about using wildcard characters.

Options

METHOD *method_name*
Specifies the method (IBMTAPE or CMSFILE).

method_parms
(See "Method parameters" on page 51 for descriptions.)

Method parameters

IBMTAPE
Indicates that a tape should be used for the input stream.

volser
Specifies the tape volume label (or SCRATCH) of the tape to be mounted.

file#
Specifies the tape file to be read as input (the label is file #0; the first stream is file #1 and so on).

RW/RO
Specifies the access: RO (read-only) or RW (read-write).

CMSFILE
Indicates that a CMS file should be used for the input stream. The CMSFILE method can be used for input streams sourced from a currently accessed CMS formatted minidisk or SFS directory.

filename
Specifies the file name of a CMS file used for input (1-8 characters; alphanumeric).

filetype
Specifies the file type of a CMS file used for input (1-8 characters; alphanumeric).

filemode
Specifies the file mode of a CMS file used for input.

EDFLOAD syntax

Restore files that were backed up to a CMS formatted minidisk.

```
►► EDFLOAD — destmode — fnmask — ftmask — fmmask — ( — METHOD — method_name ─►

   ►— method_parms ─►◄
```

Figure 20. EDFLOAD syntax

Operands

destmode
Specifies the target filemode for restore (minidisk or SFS directory).

fnmask
The wildcard selection mask for the name of the file.

ftmask
The wildcard selection mask for the type of file.

fmmask
Specifies file mode. Specify an asterisk (*) to indicate any file mode (default) or specify a file mode number.

See the *IBM z/VM CMS Commands and Utility Reference (SC24-6166)* for information about using wildcard characters.

Options

METHOD *method_name*
Specifies the method (IBMTAPE or CMSFILE).

method_parms
(See “Method parameters” on page 52 for descriptions.)

Method parameters

IBMTAPE
Indicates that a tape should be used for the input stream.

volser
Specifies the tape volume label (or SCRATCH) of the tape to be mounted.

file#
Specifies the tape file to be read as input (the label is file #0; the first stream is file #1 and so on).

RW/RO
Specifies the access: RO (read-only) or RW (read-write).

CMSFILE
Indicates that a CMS file should be used for the input stream. The CMSFILE method can be used for input streams sourced from a currently accessed CMS formatted minidisk or SFS directory.

filename
Specifies the file name of a CMS file used for input (1-8 characters; alphanumeric).

filetype
Specifies the file type of a CMS file used for input (1-8 characters; alphanumeric).

filemode
Specifies the file mode of a CMS file used for input.

FBALOAD syntax

Restore a raw FBA image backup to disk.

```
►► FBALOAD ── destvdev ── ( ── METHOD ── method_name ── method_parms ─►◄
```

Figure 21. FBALOAD syntax

Operands

destvdev
: Specifies the virtual address of the target extent.

Options

METHOD *method_name*
: Specifies the method (IBMTAPE or CMSFILE).

method_parms
: (See "Method parameters" on page 53 for descriptions.)

Method parameters

IBMTAPE
: Indicates that a tape should be used for the input stream.

volser
: Specifies the tape volume label (or SCRATCH) of the tape to be mounted.

file#
: Specifies the tape file to be read as input (the label is file #0; the first stream is file #1 and so on).

RW/RO
: Specifies the access: RO (read-only) or RW (read-write).

CMSFILE
: Indicates that a CMS file should be used for the input stream. The CMSFILE method can be used for input streams sourced from a currently accessed CMS formatted minidisk or SFS directory.

filename
: Specifies the file name of a CMS file used for input (1-8 characters; alphanumeric).

filetype
: Specifies the file type of a CMS file used for input (1-8 characters; alphanumeric).

filemode
: Specifies the file mode of a CMS file used for input.

SFSLOAD syntax

Restore files that were backed up to an SFS file space using EDFDUMP or SFSDUMP.

```
►► SFSLOAD ── destpool ── destowner ── destsg ── destpath ── pathmask ── fnmask ── ftmask ─►

   ►── fm#mask ── ( ── METHOD ── method_name ── method_parms ─►◄
```

Figure 22. SFSLOAD syntax

Operands

destpool
: Specifies the target SFS pool for the restore.

destowner

Specifies the target owner ID for the restore (1-8 characters; alphanumeric).

destsg

Specifies the target storage group (if not yet enrolled).

destpath

Specifies "." (period).

pathmask

Filter mask for original path portion of the file ID.

fnmask

Wildcard selection mask for the file name portion of the file ID.

ftmask

Wildcard selection mask for the file type portion of the file ID.

fmmask

File mode. Specify an asterisk (*) to indicate any file mode (default) or specify a file mode number (1-6).

See the *IBM z/VM CMS Commands and Utility Reference (SC24-6166)* for information about using wildcard characters.

Options

METHOD *method_name*

Specifies the method (IBMTAPE or CMSFILE).

method_parms

See "Method parameters" on page 54 for descriptions.

Method parameters

IBMTAPE

Indicates that a tape should be used for the input stream.

volser

Specifies the tape volume label (or SCRATCH) of the tape to be mounted.

file#

Specifies the tape file to be read as input (the label is file #0; the first stream is file #1 and so on).

RW/RO

Specifies the access: RO (read-only) or RW (read-write).

CMSFILE

Indicates that a CMS file should be used for the input stream. The CMSFILE method can be used for input streams sourced from a currently accessed CMS formatted minidisk or SFS directory.

filename

Specifies the file name of a CMS file used for input (1-8 characters; alphanumeric).

filetype

Specifies the file type of a CMS file used for input (1-8 characters; alphanumeric).

filemode

Specifies the file mode of a CMS file used for input.

Appendix C. Backup and restore routines

This information describes the Backup and Restore Manager data packaging routines.

REXX EXEC requirements

REXX EXEC requirements for each routine are noted for each routine where applicable.

Note: If you do not specify a variable that is required by a routine, or you do not invoke a routine from within a REXX EXEC, the routine exits with an error message.

Backup routines and input/output handlers

Backup and restore routines use input/output handlers (I/O handlers) that specify how to process input or output.

- Backup routines use input/output handlers to specify how to process the resulting output.
- Restore routines use input/output handlers to specify how to process the source backup stream.

The following input/output handlers are available for the backup and restore routines:

- Backup and restore routines use the CMSFILE input/output handler to access backup stream content that is stored in CMS files that reside on minidisk or SFS-based storage.
- Backup and restore routines use the IBMTAPE input/output handler to access backup stream content that is stored in CMS-supported tape devices, which also support a maximum data block size of 64K bytes.
- Backup and restore routines use the IBMTWIN input/output handler to access backup stream content that is stored in CMS-supported tape devices, which also support a maximum data block size of 64K bytes (used to produce output to tape "twin set" volumes).

For more information about input/output handlers, see Appendix D, “Input/Output handlers,” on page 75.

Backup routine syntax

Backup routines enable you to backup a single CKD minidisk extent, CMS formatted minidisk, or SFS file space to tape, or to a CMS file.

Required job-level variables

The backup routines (DUMPCKD, DUMPEDF, DUMPFBA, and DUMPSFS) require the following REXX variables for job-level information.

BKR_JOB_NAME
Identifies the job or process name used to generate the backup image. It is recommended that your application set this variable to the name of the application invoking the data packaging routine. (1-8 characters alphanumeric.)

BKR_JOB_INSTANCE
Identifies a specific instance of a job denoted by the variable **BKR_JOB_NAME**. Backup and Restore Manager uses this value to record the execution instance of backup or restore jobs. It is recommended that your application set this value based on the nature of the application invoking the data packaging routine. (1-8 characters alphanumeric.)

BKR_JOB_OWNER
Identifies the virtual machine that "owns" the application generating the backup image. It is recommended that your application set this variable to the user ID of the virtual machine invoking the data packaging routine. (1-8 characters alphanumeric.)

Note: CMS constraints for VM user names are applicable.

BKR_JOB_MASTER

Identifies the primary backup server. It is recommended that your application set this variable to the user ID of the virtual machine that invokes the data packaging routine, or to the application owner ID. (1-8 characters alphanumeric.)

Note: CMS constraints for VM user names are applicable.

BKR_JOB_CATALOG

Controls generation of catalog metadata by the data packaging routine as follows:

Y

Generate catalog content.

N

(The recommended setting.) Do not generate catalog content.

Note: For information about REXX and REXX environment variables, see the *z/VM REXX/VM Reference* and the *z/VM REXX/VM User's Guide*. For information about CMS, see the *z/VM CMS User's Guide*.

DUMPCKD

Use DUMPCKD to package a track-image dump of a CKD DASD extent (minidisk or full-volume) into a backup stream. Output storage media is determined by the selection of a particular input/output handler name.

The DUMPCKD routine collects data from a target CKD DASD device on a track-by-track basis. The actual format of the DASD extent is irrelevant. The format allows applications to package CKD DASD contents for both native VM data and non-VM DASD such as z/OS, VSE, or Linux on zSeries.

Note: Usability of the backup stream can depend on whether the contents of the CKD DASD extent change during data capture. The application must determine whether the source data is quiesced prior to data packaging.

Syntax

```
▶▶─ DUMPCKD ── ##REXX ─▶◀
```

Figure 23. DUMPCKD syntax

where *##REXX* acknowledges that the DUMPCKD routine is invoked from within a REXX EXEC.

Note: DUMPCKD requires minidisk address 3F0 to be available.

DUMPCKD-specific variables

In addition to the job-level variables that are described in “Required job-level variables” on page 55, the DUMPCKD routine requires the following REXX variables:

BKR_CKD_OWNER

A valid VM user ID that identifies the owner of the target minidisk (1-8 characters; alphanumeric).

BKR_CKD_VDEV

A valid virtual device address that identifies the minidisk belonging to the owner that is specified by **BKR_CKD_OWNER** that is being targeted for packaging.

BKR_CKD_TOKEN

1-8 character alphanumeric token that can be used as a means to organize catalog metadata. Typically this value is set to numeric date such as: 20060124 (01/24/2006).

BKR_CKD_METHOD

Identifies the output handler that should be invoked to process resulting output. Specify one of the following options:

CMSFILE
Output to a CMS file.

IBMTAPE
Output to a single tape volume.

IBMTWIN
Output to tape "twin set" volumes.

Usage notes

To use DUMPCKD, applications are expected to have one of the following:

- OPTION LNKNOPAS in effect in the CP directory, or
- Prepared to provide an SR or RR LINK password when the target CKD DASD extent is LINKed by DUMPCKD

If you use &SYSRES or DEVNO MDISK definitions, all worker task virtual machines require additional privileges. Add OPTION DEVINFO in their CP directory entries or equivalent RACF® or other ESM authorization if an ESM such as RACF is used.

Messages

Table 22 on page 57 lists the messages generated by DUMPCKD.

Table 22. DUMPCKD messages

Message number	**Message text**
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8002	REXX environment must be active.
8003	Unrecognized output handler *handler* specified.
8004	Output handler *handler* has been disabled.
8005	Return code *rc* from output handler initialization.
8006	Return code *rc* from catalog data initialization.
8176	Output handler refused CKD CTNRDATA, return code *rc*.
8177	Catalog handler refused CKD CTNRDATA, return code *rc*.
8178	BKRRTRK return code *rc* attempting to read track *track*.

DUMPEDF

Use DUMPEDF to package files from an Enhanced Disk Format (EDF) CMS minidisk into a backup stream. Output storage media is determined by the selection of a particular input/output handler name.

The DUMPEDF routine packages minidisk attributes (EDF block size, minidisk label, and other information stored in the ADT data structure) and files that pass wildcard filtering, and file-level attributes as specified.

Note: DUMPEDF tolerates invalid filename or file type characters in an File Status Table (FST) entry.

Syntax

```
►► DUMPEDF ── ##REXX ─►◄
```

Figure 24. DUMPEDF syntax

where *##REXX* acknowledges that the DUMPEDF routine is invoked from within a REXX EXEC.

Note: DUMPEDF requires minidisk address 3F0 and filemode letter Z to be available.

DUMPEDF-specific variables

In addition to the job-level variables described in "Required job-level variables" on page 55, the DUMPEDF routine requires the following REXX variables:

BKR_EDF_OWNER
A valid VM user ID that identifies the owner of the target minidisk (1-8 characters; alphanumeric).

BKR_EDF_VDEV
A valid virtual device address that identifies the minidisk belonging to the owner that is specified by `BKR_EDF_OWNER` that is being targeted for packaging.

BKR_EDF_FNMASK
A wildcard expression for a filename used to filter selection of files from the target minidisk (1-8 characters; alphanumeric).

BKR_EDF_FTMASK
A wildcard expression for a filetype used to filter selection of files from the target minidisk (1-8 characters; alphanumeric).

BKR_EDF_FMMASK
An asterisk (*) or a single integer value (0-6) used to filter selection of files from the target minidisk.

BKR_EDF_TOKEN
1-8 character alphanumeric token that can be used as a means to organize catalog metadata. Typically this value is set to numeric date such as: 20050124 (01/24/2005).

BKR_EDF_INCR_TOGGLE
Determines whether or not the target minidisk should be processed for an incremental backup or a full backup. Specify one of the following options:

Y
Perform incremental backup processing.

N
(The recommended setting.) Perform full backup processing.

BKR_EDF_PRELINK
Specify one of the following:

Y
Indicates that the target minidisk has already been LINKed and ACCESS as filemode Z by the calling routine.

N
Indicates the DUMPEDF routine attempts to LINK the target minidisk at address 3F0 and, if successful, ACCESS the target as filemode Z with the MODE0 option in effect on the **ACCESS** command.

BKR_EDF_METHOD
Identifies the output handler that should be invoked to process resulting output. Specify one of the following options:

CMSFILE
Output to a CMS file.

IBMTAPE
Output to a single tape volume.

IBMTWIN
Output to tape "twin set" volumes.

Usage notes

Applications are expected to either:

- Have OPTION LNKNOPAS in effect in the CP directory. Set **BKR_EDF_PRELINK** to N and allow DUMPEDF to handle LINK and ACCESS of the minidisk internally, perform packaging operations, and then RELEASE and DETACH the target.
- LINK the target minidisk and ACCESS the target minidisk as filemode Z before you invoke DUMPEDF. Invoke DUMPEDF with **BKR_EDF_PRELINK** set to Y and then RELEASE and DETACH the target minidisk when DUMPEDF completes processing.

If you use &SYSRES or DEVNO MDISK definitions, all worker task virtual machines require additional privileges. Add OPTION DEVINFO in their CP directory entries or equivalent RACF or other ESM authorization if an ESM such as RACF is used.

Messages

Table 23 on page 59 lists the messages generated by DUMPEDF.

Table 23. DUMPEDF messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8002	REXX environment must be active.
8003	Unrecognized output handler *handler* specified.
8004	Output handler *handler* has been disabled.
8005	Return code *rc* from output handler initialization.
8006	Return code *rc* from catalog data initialization.
8157	CP return code *rc* from command *command*.
8158	CP response: *response*.
8159	CMS return code *rc* from command ACCESS 3F0 Z (MODE0.
8160	Return code *rc* from VMUDQ inquiry.
8161	Incomprehensible response from VMUDQ.
8162	Reply length was *n* bytes.
8163	ADT search failure.
8164	Source minidisk is not in EDF format.
8165	Output handler refused EDF CTNRDATA, return code *rc*.
8166	Catalog handler refused EDF CTNRDATA, return code *rc*.
8167	FST size inconsistency (test #1).
8168	FST size inconsistency (test #2).
8169	Could not locate any FSTs to process.
8170	DMSEXIST return code *rc*, reason code *reason_code*.
8171	FILEHEAD return code *rc*.
8172	Catalog FILEHEAD return code *rc*.
8173	Unrecognized or unsupported EDF block size.
8174	FILEDATA return code *rc*.
8175	FILEEND return code *rc*.

DUMPFBA

The DUMPFBA routine collects data from a target FBA DASD device. Use DUMPFBA to package a track-image dump of a FBA DASD extent, either minidisk or full volume, into a backup stream. Output storage media is determined by the selection of a particular input/output handler name.

Syntax

```
►► DUMPFBA — ##REXX ►◄
```

Figure 25. DUMPFBA syntax

where *##REXX* acknowledges that the DUMPFBA routine is being invoked from within a REXX EXEC.

DUMPFBA-specific variables

In addition to the job-level variables described in “Required job-level variables” on page 55, the DUMPFBA routine requires the following REXX variables:

BKR_FBA_OWNER
A valid VM user ID that identifies the owner of the target minidisk (1-8 characters; alphanumeric).

BKR_FBA_VDEV
A valid virtual device address that identifies the minidisk that belongs to the owner that is specified by **BKR_FBA_OWNER** that is targeted for packaging.

BKR_FBA_METHOD
Identifies the output handler to invoke to process resulting output. Specify one of the following options:

CMSFILE
Output to a CMS file.

IBMTAPE
Output to a single tape volume.

IBMTWIN
Output to tape "twin set" volumes.

BKR_FBA_TOKEN
1-8 character alphanumeric token that can be used as a means to organize catalog metadata. Typically this value is set to numeric date such as: 20060124 (01/24/2006).

BKR_FBA_PRELINK
Specify one of the following options:

Y
The target minidisk was LINKed and ACCESS as filemode Z by the calling routine.

N
The DUMPFBA routine attempts to LINK the target minidisk at address 3F0 and, if successful, ACCESS the target as filemode Z with the MODE0 option in effect on the **ACCESS** command.

Messages

Table 24 on page 60 lists the messages generated by DUMPFBA.

Table 24. DUMPFBA messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8002	REXX environment must be active.
8003	Unrecognized output handler *handler* specified.

Table 24. DUMPFBA messages (continued)

Message number	Message text
8004	Output handler *handler* has been disabled.
8005	Return code *rc* from output handler initialization.
8006	Return code *rc* from catalog data initialization.
9124	Output handler refused FBA CTNRDATA, return code *rc*.
9125	Catalog handler refused FBA CTNRDATA, return code *rc*.
9126	BKRFBA return code *rc* trying to read track *track*.

DUMPSFS

The DUMPSFS routine packages objects from a SFS file space into a backup stream. Use DUMPSFS to package SFS file space attributes, base files, aliases, external objects, and directories as well as authorizations that results from use of the **GRANT** command. Output storage media is determined by the selection of a particular input/output handler name.

Syntax

```
►►— DUMPSFS —— ##REXX —►◄
```

Figure 26. DUMPSFS syntax

where *##REXX* acknowledges that the DUMPSFS routine is being invoked from within a REXX EXEC.

Note: DUMPSFS requires minidisk address 3F0 to be available.

DUMPSFS-specific variables

In addition to the job-level variables described in "Required job-level variables" on page 55, the DUMPSFS routine requires the following REXX variables:

BKR_SFS_POOL
A valid SFS file pool name that identifies the file pool which contains the target file space.

BKR_SFS_OWNER
A valid SFS file space name that identifies the file space to be packaged.

BKR_SFS_DIRPATH
A wildcard expression used to filter file space contents by directory path.

BKR_SFS_FNMASK
A wildcard expression for a filename used to filter selection of files from the target minidisk (1-8 characters; alphanumeric).

BKR_SFS_FTMASK
A wildcard expression for a file type used to filter selection of files from the target minidisk (1-8 characters; alphanumeric).

BKR_SFS_FMMASK
An asterisk (*) or a single integer value (0-6) used to filter selection of files from the target minidisk.

BKR_SFS_VERBOSE
Controls volume of informational messages issued by DUMPSFS. Specify one of the following options:

Y
Provide object-by-object description of runtime actions (more verbose output).

N
Provide "quiet" behavior (less verbose output).

BKR_SFS_TOKEN

1-8 character alphanumeric token that can be used as a means to organize catalog metadata. Typically this value is set to numeric date such as: 20050124 (01/24/2005).

BKR_SFS_INCR_TOGGLE

Determines whether the target minidisk should be processed for an incremental backup or a full backup. Specify one of the following options:

Y

Perform incremental backup processing.

N

(Recommended setting) Perform full backup processing.

BKR_SFS_METHOD

Identifies the output handler that should be invoked to process resulting output. Specify one of the following options:

CMSFILE

Output for a CMS file.

IBMTAPE

Output for a single tape volume.

IBMTWIN

Output for tape "twin set" volumes.

Usage notes

Customer applications are expected to have SFS ADMIN authority in the target file pool. Non-privileged clients can run DUMPSFS to target file spaces to which they were given access, but comprehensive extraction of all attributes from the targeted file space is not guaranteed.

Messages

Table 25 on page 62 lists the messages generated by DUMPSFS.

Table 25. DUMPSFS messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8002	REXX environment must be active.
8004	Output handler *handler* has been disabled.
8005	Return code *rc* from output handler initialization.
8006	Return code *rc* from catalog data initialization.
8007	DMSDISFS return code *rc*, reason code *reason_code*.
8008	DMSQLIMU return code *rc*, reason code *reason_code*.
8009	CTNRDATA(FILESPACE) return code *rc*.
8010	Catalog CTNRDATA(FILESPACE) return code *rc*.
8011	DMSOPDIR return code *rc*, reason code *reason_code*.
8012	DMSGETDA return code *rc*, reason code *reason_code*.
8013	Base file: *dirname/filename*.
8014	DMSEXIST (base file) return code *rc*, reason code *reason_code*.
8015	INCRTEST fatal error, return code *rc*.

Table 25. DUMPSFS messages (continued)

Message number	**Message text**
8016	INCRTEST return code *rc*; incremental processing disabled.
8017	DMSOPDBK return code *rc*, reason code *reason_code*; Target file was *filename filetype*.
8018	Base file is active in SMS; reason code *rc*.
8019	FILEHEAD (base file) return code *rc*.
8020	SFS FILEHEAD catalog call gave return code *rc*.
8021	DMSRDDBK return code *rc*, reason code *reason_code*.
8022	FILEDATA (base file) return code *rc*.
8023	FILEEND (base file) return code *rc*.
8024	GETAUTH (base file) return code *rc* ; R1 = *parmaddr*.
8025	SFSATTR (base file) return code *rc*.
8026	DMSCLDBK return code *rc*, reason code *reason_code*.
8027	DMSEXIST (alias) return code *rc*, reason code *reason_code*.
8028	Target was *dirname* /*filename filetype*.
8029	Alias: *dirname*/*filename filetype*.
8030	GETBASE failed to resolve base object.
8031	---Base object: *sfs_directory_path*/*filename filetype*.
8032	---Owner ID: *id*.
8033	FILEHEAD (alias) return code *rc*.
8034	Catalog FILEHEAD (alias) return code *rc*.
8035	FILEDATA (alias) return code *rc*.
8036	FILEEND (alias) return code *rc*.
8037	GETAUTH (alias) return code *rc*.
8038	SFSATTR (alias) return code *rc*.
8039	Dropping erased alias *directory_path*/*filename filetype*.
8040	Dropping revoked alias *directory_path*/*filename filetype*.
8041	---SFS Directory: *directory*.
8042	DMSEXIST (directory) return code *rc*, reason code *reason_code*.
8043	CTNRDATA (directory) return code *rc*.
8044	Catalog CTNRDATA (directory) return code *rc*.
8045	GETAUTH (directory) return code *rc* , R1 = *parmaddr*.
8046	SFSATTR (directory) return code *rc*.
8047	DMSEXIST (external object) return code *rc*, reason code *reason_code*.
8048	External Object: *directory_path*/*filename filetype*.
8049	DMSQOBJ return code *rc*, reason code *reason_code*.

Table 25. DUMPSFS messages (continued)

Message number	Message text
8050	--- Refers to: *object_name*.
8051	--- Xobj Type: *type*.
8052	FILEHEAD (external object) return code *rc*.
8053	Catalog FILEHEAD (external object) return code *rc*.
8054	FILEDATA (external object) return code *rc*.
8055	FILEEND (external object) return code *rc*.
8056	DMSGETDA status 7 with *dirname/filename filetype*.
8057	DMSGETDA status 8 with *dirname/filename filetype*.
8058	DMSCLDIR return code *rc*, reason code *reason_code*.
8059	DMSENAFS return code *rc*, reason code *reason_code*.
8060	Return code *rc* from output handler termination.
8061	Return code *rc* from catalog data termination.

Restore routine syntax

Restore routines enable you to restore data. The following restore routines are provided with Backup and Restore Manager.

Required job level variables

Note: The restore routines LOADCKD, LOADDDL, LOADEDF, LOADFBA, and LOADSFS do not require REXX variables for job-level information.

LOADCKD

The LOADCKD routine restores a track-image dump of a CKD DASD extent, (minidisk or full-volume) from a backup stream.

The LOADCKD routine restores a track-image dump of a CKD DASD extent (minidisk or full-volume) from a backup stream. Input storage media is determined by the selection of a particular input/output handler name.

The restore target must be of the same device type, and the target extent must consist of at least as many cylinders as the source extent. The target extent can be larger than the source, but capacity beyond the size of the source extent might, or might not, be usable depending on the data you are restoring.

Note: The LOADCKD routine does not reserve a specific minidisk address or filemode. The target minidisk is specified as part of the REXX environment. The LOADCKD routine extracts minidisk device type and capacity information about the restore target, and compares it against the source extent.

Syntax

```
▶▶— LOADCKD —— ##REXX —▶◀
```

Figure 27. LOADCKD syntax

where *##REXX* acknowledges that the DUMPSFS routine is invoked from within a REXX EXEC.

LOADCKD-specific variables

The LOADCKD routine requires the following REXX variables:

BKR_CKD_TARGET_VDEV

A valid minidisk address that identifies the virtual device to use as the restore target.

Note: LINK the minidisk before you invoke LOADCKD (required).

BKR_CKD_VERBOSE

Controls the volume of informational messages that LOADCKD issues. Specify one of the following options:

Y

Issue console messages that identify the job that generated the backup image and provide source media descriptions.

N

Produce less verbose output.

BKR_CKD_INPUT_METHOD

Identifies the input handler to invoke to process the source backup stream. Specify one of the following options:

CMSFILE

A CMS file.

IBMTAPE

A single tape volume.

Usage notes

Client applications that invoke LOADCKD do not require privileges beyond CP class G and read/write access to the target DASD extent.

Messages

Table 26 on page 65 lists the messages that are generated by LOADCKD.

Table 26. LOADCKD messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8002	REXX environment must be active.
8003	Unrecognized output handler *handler* specified.
8004	Output handler *handler* has been disabled.
8005	Return code *rc* from output handler initialization.
8006	Return code *rc* from catalog data initialization.
8116	Unrecognized input handler *handler* specified.
8117	Initialization of *handler* failed; return code *rc*, reason code *reason_code*.
8118	Input handler *handler* return code *rc*, reason *reason_code* during GETDATA.
8119	Input stream sequencing error.
8120	Start of buffer: *buffer_start*.
8121	Backup image generated by job *job*, instance *instance*.
8122	FHCTYPE / FHFTYPE mismatch.

Table 26. LOADCKD messages (continued)

Message number	**Message text**
8123	Starting restore of raw CKD dump; track range *n1* - *n2*.
8124	FHCTYPE is not CKD.
8125	Input stream sequencing error; FILEHEAD found before CTNRDATA.
8126	Return code *rc* from BKRWTRK.
8127	Unable to continue; read *n* bytes, track image length is *length*.
8128	Source data is a CKD image dump for *owner addr*.
8129	Target extent size (*n* cyls) is compatible with source data.
8130	CTNRDATA is not flagged as CKD.
8131	Invalid device address.
8132	BKRD210 return code *rc* during extract of target description.
8133	Source and target device types do not match.
8134	Target extent has too few cylinders to contain source image.
8135	Operation complete; *n* tracks restored to target extent.
8136	Input handler *handler* return code *rc*, reason code *reason* on TERMINAT.

LOADDDL

The LOADDDL routine restores content directly to a spool destination.

Syntax

```
►►─ LOADDDL ── ##REXX ─►◄
```

where *##REXX* acknowledges that the LOADDDL routine is invoked from within a REXX EXEC.

Figure 28. LOADDDL syntax

LOADDDL-specific variables

The LOADDDL routine requires the following REXX variables:

BKR_DDL_DEST_USER
The destination user for the restored files.

BKR_DDL_DEST_NODE
The destination RSCS/NJE node name. A dash (-) character specifies the local node.

BKR_DDL_LOCAL_NODE
The RSCS/NJE node name of the local system.

BKR_DDL_RSCSID
The virtual machine name of the local RSCS service virtual machine.

BKR_DDL_ORIGID
The user name to insert into the NETDATA output as the user-of-origin for the files to restore.

BKR_DDL_PAMASK
A wildcard expression to filter restore of SFS data by an SFS directory path.

BKR_DDL_FNMASK
1-8 character regular expression. The LOADDDL routine filters the source backup stream with the **BKR_DDL_FNMASK** and the **BKR_DDL_FTMASK** settings to select only filename and filetype combinations that match the expression. Specify an asterisk (*') to accept all files.

BKR_DDL_FTMASK
1-8 alphanumeric wildcard expression for a filetype to filter the selection of files from the target minidisk.

BKR_DDL_FMMASK
1-byte regular expression. Specify an asterisk (*') or a percent sign ('%) to select all filemode numbers. Specify a valid filemode number (0...6) to select a subset from the source stream by filemode number.

BKR_DDL_INPUT_METHOD
The input handler that is invoked to process the source backup stream. Specify one of the following options:

CMSFILE
Input for a CMS file.

IBMTAPE
Input for a single tape volume.

For more information, see Appendix D, "Input/Output handlers," on page 75.

BKR_DDL_RECORD_LIMIT
The maximum number of spool output records to generate per restored file.

BKR_DDL_SPOOL_CLASS
The CP pool class to use for restored output.

BKR_DDL_VERBOSE
Controls volume of informational messages that are issued by LOADDDL. Specify one of the following options:

Y
Issue console messages that identify the job that generated the backup image and provide source media descriptions.

N
Produce less verbose output.

Messages

Table 27 on page 67 lists the messages that are generated by LOADDDL.

Table 27. LOADDDL messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8002	REXX environment must be active.
8063	IJPARML format inconsistency.
8080	FHPARML inconsistency encountered.
8116	Unrecognized input handler *handler* specified.
8117	Initialization of *handler* failed; return code *rc*, reason code *reason_code*.
8118	Input handler *handler* return code *rc*, reason *reason_code* during GETDATA.
8119	Input stream sequencing error.
8120	Start buffer: *buffer_start*.

Table 27. LOADDDL messages (continued)

Message number	Message text
8121	Backup image generated by job *job*, instance *instance*.
8122	FHCTYPE / FHFTYPE mismatch.
8136	Input handler *handler* return code *rc*, reason code *reason_code* on TERMINATE.
8138	Found *filename filetype*, owner *owner*, source *source*.
8142	Impossible result returned from block size calculation.
8153	Dropping SFS ALIAS definition; not applicable for EDF target.
8154	Dropping SFS AUTH definition; not applicable for EDF target.
8155	Dropping SFS External Object; not applicable for EDF target.
9111	Invalid numeric character in BKR_DDL_RECORD_LIMIT ("*value*").
9112	Return code *rc* from NETDATA output handler during WRITE operation.
9113	Return code *rc* from NETDATA output handler during TERMINATE operation.
9114	Restore complete; *nn* files delivered to spool destination.
9117	Spool output record limit reached for file; output has been flushed. Processing continues with the next file.

LOADEDF

The LOADEDF routine can restore either a file-level dump of a CMS formatted minidisk, produced by DUMPEDF, to a CMS formatted minidisk target, or it can restore base files from an SFS file space dump produced by DUMPSFS to a CMS formatted minidisk target.

The LOADEDF routine extracts minidisk device type and capacity information about the restore target, and compares it against the source extent. The restore target must be of the same device type, and the target extent must consist of at least as many cylinders as the source extent. The target extent can be larger than the source, but capacity beyond the size of the source extent might, or might not, be usable depending on the data you are restoring.

LOADEDF expects the calling application to ensure that the restore target has sufficient capacity to contain files you are restoring from the source backup stream. The restore target minidisk can be of any CMS-supported DASD type when the EDF blocksize compatibility requirement is met. The restore target can also be an SFS directory if the source data is derived from an SFS file space or a 4K-formatted CMS formatted minidisk.

Notes:

1. Using LOADEDF to restore an SFS backup stream to an SFS target only restores base files. Directories, aliases, external objects, and permissions are not re-created by LOADEDF.
2. The LOADEDF routine does not reserve a specific minidisk address or file mode. The target file mode is specified as part of the REXX environment.

Syntax

```
►► LOADEDF —— ##REXX ►◄
```

where *##REXX* acknowledges that the LOADEDF routine is invoked from within a REXX EXEC.

Figure 29. LOADEDF syntax

Use LOADEDF to restore one of the following items:

- A file-level dump of a CMS formatted minidisk, produced by DUMPEDF, to a CMS formatted minidisk target.
- A base file from an SFS file space dump that is produced by DUMPSFS to a CMS formatted minidisk target.

LOADEDF-specific variables

The LOADEDF routine requires these REXX variables:

BKR_EDF_ACCESS_MODE
The file mode of the LOADEDF restore target (a file mode letter for a currently accessed minidisk or SFS directory).

BKR_EDF_FNMASK
1-8 character regular expression. The LOADEDF routine filters the source backup stream using the **BKR_EDF_FNMASK** and **BKR_EDF_FTMASK** settings to select only file name and file type combinations that match the expression. Specify an asterisk (*') to accept all files.

BKR_EDF_FTMASK
A wildcard expression for a filetype that is used to filter selection of files from the target minidisk (1-8 characters; alphanumeric).

BKR_EDF_FMMASK
1-byte regular expression. Specify an asterisk (*') or a percent sign ('%) to select all file mode numbers. Specify a valid file mode number (0..6) to select a subset from the source stream by file mode number.

BKR_EDF_VERBOSE
Controls volume of informational messages that are issued by LOADEDF. Specify one of the following values:

Y
Issue console messages that identify the job that generated the backup image and provide source media descriptions.

N
Produce less verbose output.

BKR_EDF_INPUT_METHOD
The input handler that is invoked to process the source backup stream. Specify one of the following values:

CMSFILE
Input for a CMS file.

IBMTAPE
Input for a single tape volume.

For more information, see Appendix D, “Input/Output handlers,” on page 75.

Usage notes

LOADEDF requires that the target minidisk already be initialized by the CMS **FORMAT** command. The CMS formatted minidisk blocksize must match the block size of the source backup stream. When you are restoring SFS base files from a backup stream that is generated by DUMPSFS, the target minidisk must be formatted with a block size of 4096 (4K) bytes.

Client applications that invoke LOADEDF require no extraordinary privileges beyond CP class G and read/write access to the target DASD extent.

Messages

Table 28 on page 70 lists the messages that are generated by LOADEDF.

Table 28. LOADEDF messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8002	REXX environment must be active.
8003	Unrecognized output handler *handler* specified.
8004	Output handler *handler* was disabled.
8005	Return code *rc* from output handler initialization.
8006	Return code *rc* from catalog data initialization.
8153	Dropping SFS ALIAS definition; not applicable for EDF target.
8154	Dropping SFS AUTH definition; not applicable for EDF target.
8155	Dropping SFS External Object; not applicable for EDF target.
8156	Operation complete; restored *n* files to target.

LOADFBA

The LOADFBA routine restores a raw FBA backup data stream. The restore target must be of the same device type.

Syntax

```
►►─ LOADFBA ── ##REXX ─►◄
```

where *##REXX* acknowledges that the LOADFBA routine is invoked from within a REXX EXEC.

Figure 30. LOADFBA syntax

LOADFBA-specific variables

The LOADFBA routine requires these REXX variables:

LOAD_FBA_TARGET
A valid minidisk address that identifies the virtual device to use as the restore target.

Note: LINK the minidisk before you invoke LOADFBA (required).

LOAD_FBA_INPUT_METHOD
The input handler that is invoked to process the source backup stream. Specify one of the following options:

CMSFILE
A CMS file.

IBMTAPE
A single tape volume.

LOAD_FBA_VERBOSE
Controls volume of informational messages that are issued by LOADFBA. Specify one of the following options:

Y
Issue console messages that identify the job that generated the backup image and provide source media descriptions.

N
Produce less verbose output.

Messages

Table 29 on page 71 lists the messages that are generated by LOADFBA.

Table 29. LOADFBA messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8002	REXX environment must be active.
8003	Unrecognized output handler *handler* specified.
8004	Output handler *handler* has been disabled.
8005	Return code *rc* from output handler initialization.
8006	Return code *rc* from catalog data initialization.
9118	Starting restore of raw FBA dump; block range *n1* - *n2*.
9119	Source data is a FBA image dump for *n1 n2*.
9120	Operation complete; *n* blocks restored to target extent.
9121	Target extent size (*n* blks) is compatible with source data.
9122	Target extent has too few cylinders to contain source image.
9123	CTNRDATA is not flagged as FBA.

LOADSFS

The LOADSFS routine restores the contents and attributes of an SFS file space from a previously generated DUMPSFS backup stream.

Note: LOADSFS does not reserve minidisk addresses or filemode letters. The target file space is specified as part of the REXX environment. If the client has ADMIN privileges to the target file pool, and the target file space is not enrolled, the file space is re-enrolled with the original storage group with the storage limits in place when the backup stream was created. If the client does not have ADMIN privileges, the restore operation fails when LOADSFS attempts to re-enroll the target file space.

LOADSFS requires that the client has ADMIN privileges in the file pool to which the restore operation is being performed, or that the client owns the file space to which the data is being restored. LOADSFS re-creates the original directory hierarchy that is contained in the backup stream, and restores base files, directories, aliases, external objects, and third-party access.

Syntax

```
▶▶─ LOADSFS ── ##REXX ─▶◀
```

where *##REXX* acknowledges that the LOADSFS routine is invoked from within a REXX EXEC.

Figure 31. LOADSFS syntax

LOADSFS-specific variables

The LOADSFS routine requires the following REXX variables:

BKR_SFS_APOOL
The target file pool for the restore operation. The file pool must be active at the time of the restore operation.

BKR_SFS_AOWNER
The target file space for the restore operation. If the client has ADMIN privileges and the file space is not already enrolled, the file space is re-created with the limits in effect at the time that backup data was created.

BKR_SFS_ALTSG
A valid storage group number. If the target file space is not already enrolled, it is enrolled in the specified storage group.

BKR_SFS_PATHMASK
1-8 character wildcard expression to filter the input stream by directory path.

BKR_SFS_FNMASK
1-8 character regular expression. The LOADSFS routine filters the source backup stream using the **BKR_SFS_FNMASK** and **BKR_SFS_FTMASK** settings to select only file name and file type combinations that match the expression. Specify an asterisk (*') to accept all files.

BKR_SFS_FTMASK
A wildcard expression for a file type to filter selection of files from the target minidisk (1-8 characters; alphanumeric).

BKR_SFS_FMMASK
1-byte regular expression. Specify an asterisk (*') or a percent sign ('%) to select all filemode numbers. Specify a valid filemode number (0..6) to select a subset from the source stream by filemode number.

BKR_SFS_VERBOSE
Controls the volume of informational messages LOADSFS issues. Specify one of the following options:

Y
Issue console messages that identify the job that generated the backup image and provides source media descriptions.

N
Provide less verbose output.

BKR_SFS_INPUT_METHOD
The input handler that is invoked to process the source backup stream. Specify one of the following options:

CMSFILE
A CMS file.

IBMTAPE
A single tape volume.

For more information, see Appendix D, “Input/Output handlers,” on page 75

Usage notes

In some cases, if the source backup stream includes ALIAS definitions, it might be necessary to perform a "two pass" restore operation. In this situation, the first restore operation re-creates directories and base objects, but can encounter warnings when it attempts to re-create ALIAS definitions that relate to base objects that have not been restored yet. This situation occurs when CSL routines that are used during creation of the backup stream receive ALIAS definitions before receiving the base objects to which aliases relate. A second restore operation with identical parameters skips restoration of pre-existing objects, but is then able to re-create ALIAS definitions after the necessary base files are in place.

Messages

Table 30 on page 73 lists the messages that are generated by LOADSFS.

Table 30. LOADSFS messages

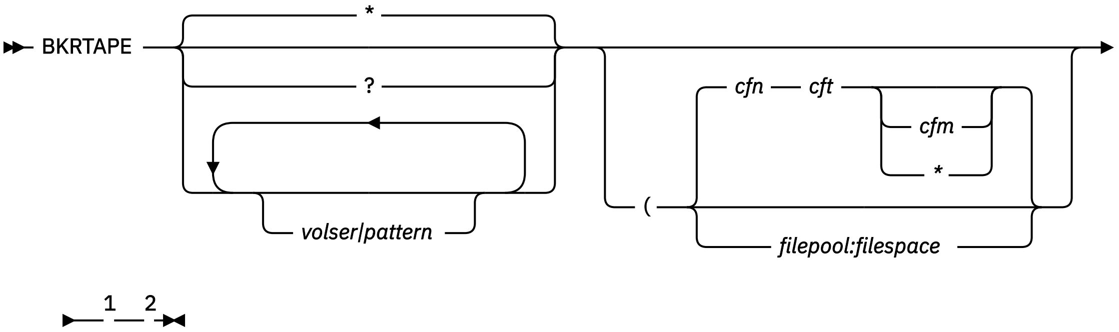
Message number	**Message text**
8001	BKRGETRX return code *rc* while attempting to fetch *variable*.
8002	REXX environment must be active.
8003	Unrecognized output handler *handler* specified.
8004	Output handler *handler* was disabled.
8005	Return code *rc* from output handler initialization.
8006	Return code *rc* from catalog data initialization.
8137	Invalid numeric data in target storage group number "*n*".
8138	Found *filename filetype*, owner *owner*, source *source*.
8139	File skipped; "*file*" already exists on restore destination.
8140	DMSOPDBK (edf) return code *rc*, reason code *reason_code*.
8141	DMSOPDBK (sfs) return code *rc*, reason code *reason_code*.
8142	Impossible result returned from block size calculation.
8143	DMSWRDBK return code *rc*, reason code *reason_code*.
8144	DMSCLDBK return code *rc*, reason code *reason_code*.
8145	File space *filespace* is already enrolled; present limits will apply.
8146	File space enrollment failed, return code *rc* reason code *reason_code*.
8147	DEFSFDIR return code *rc*, reason code *reason_code*.
8148	ALIAS skipped; "*filename filetype directory_path*" already exists.
8149	ALIAS not restored; DEFALIAS return code *rc*, reason code *reason_code*.
8150	DEFAUTH return code *rc*, reason code *reason_code*.
8151	EXTERNAL OBJECT skipped; "*filename filetype directory_path*" already exists.
8152	DEFEXOBJ return code *rc*, reason code *reason_code*.

Appendix D. Input/Output handlers

Input and output handlers are used by backup and restore routines.

BKRTAPE

The **BKRTAPE** command reads a Backup and Restore Manager catalog and reports, for each tape that matches the selection criteria, the associated job and instance. The default is to report all tapes, but one or more tape volsers or patterns can be used to restrict the output.



Notes:

[1] If you invoke BKRTAPE without alternate configuration parameters (*cfn*, *cft*, *cfm*) or an SFS catalog (*filepool*:*filespace*), it searches for the default configuration file `BKRSYSTM CONFIG` on any accessed filemode.

[2] If you invoke BKRTAPE with an alternate configuration file parameter (*cfn*, *cft*), *cfm* defaults to * if not specified.

Figure 32. BKRTAPE command syntax

Authorization

BKRTAPE requires SFS administrator authorization to the Backup and Restore Manager catalog.

Operands

?

Types a short description of BKRTAPE operation and syntax.

volser|pattern

Restricts BKRTAPE to reporting only on a specific tape (or tapes) rather than the default of all tapes.

If the operand ends in an asterisk (*), it represents a set of tapes that begin with a common character string. For example, `BKR0*` represents all tapes whose volume serials begin with the characters `BKR0`.

Operands that do not end in an asterisk represent a single volume serial.

You can specify as many volsers or patterns as you want, and the result will be a display of all tapes that meet any of the criteria.

cfn

The file name of an alternative configuration file.

cft

The file type of an alternative configuration file.

cfm
The file mode of an alternative configuration file. If you do not specify a mode, the default is * (any accessed file mode).

filepool
The SFS catalog file pool to browse for backup jobs, instead of obtaining the file pool from a backup configuration file.

Note: You must specify *filepool* with *filespace*.

filespace
The SFS file space to browse for backup jobs, instead of obtaining the file space from a backup configuration file.

Note: You must specify *filespace* with *filepool*.

Usage notes

Because a tape can, and usually will, be part of a "set" of tapes that are associated with the same job and instance, BKRTAPE output is structured to group tapes that are associated with a specific job and instance together. However, tape selection criteria take precedence. Therefore, if some tapes in a set meet the criteria and others do not, only the tapes that meet the criteria are shown.

CMSFILE

Backup and restore routines use the CMSFILE input/output handler to access backup stream content that is stored in CMS files that reside on either minidisk or SFS-based storage. The backup stream files consist of a combination of client data that is contained in the backup, and metadata that contains information about the task that created the backup stream.

Note: Invoke the CMSFILE input/output handler for either output (creation of a new backup stream file) or input (access of an existing backup stream file).

Variables required for input processing

When you invoke CMSFILE for input processing by the LOADEDF, LOADDDL, LOADFBA, LOADCKD, or LOADSFS routines, the following variables must be present in the REXX environment:

BKR_INP_EDF_FN
The file name of the input file (a valid CMS file name). The file must be present on a minidisk or directory that was made available to the application by the **ACCESS** command. 1-8 characters.

BKR_INP_EDF_FT
The file type of the input file (a valid CMS file type). The file must be present on a minidisk or directory that was made available to the application by the **ACCESS** command. 1-8 characters.

BKR_INP_EDF_FM
A valid CMS file mode. This value consists of a single alphabetic character (A..Z) that represents a currently accessed minidisk or directory, or a letter and number such as A2, that represents the file mode of the desired input file. 1-2 characters.

Variables required for output processing

When you invoke CMSFILE for output processing by the DUMPEDF, DUMPCKD, DUMPFBA, or DUMPSFS routines, the following variables must be present in the REXX environment:

BKR_OUT_EDF_FN
The filename of a new file that is created to contain the backup stream content that is generated by the associated dump routine (a valid CMS file name). 1-8 characters.

BKR_OUT_EDF_FT
The filetype of a new file that is created to contain the backup stream content that is generated by the associated dump routine (a valid CMS file type). 1-8 characters.

BKR_OUT_EDF_FM

A valid CMS filemode. This value consists of one alphabetic character (A..Z) that identifies a previously accessed minidisk or directory, or a letter and number concatenation such as : A2 that identifies a file mode letter and number for the resulting output file. 1-2 characters; alphanumeric.

BKR_OUT_EDF_REBLOCK

Controls input/output reblocking. Specify one of the following options:

Y

Data records are reblocked to 64K. If you configured UDPEs (User Data Processing Exits), they are invoked as described in the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*.

Note: Input/output reblocking services add information at the beginning of each physical output record. The record header information is used for the management of UDPEs and data reblocking. Unless you configured a data compression UDPE such as BKREXT3A, it can cause the resulting output file to consume more disk space than the equivalent non-re-blocked version. Do not use data reblocking for CMSFILE output unless you also employ a data compression UDPE.

N

Data records are not reblocked (default).

BKR_OUT_EDF_VERBOSE

Controls the number of console messages that are generated during processing. Specify one of the following options:

Y

Generate informational console messages during processing.

N

Suppress non-critical messages.

Usage notes

When invoked in output mode, CMSFILE requires that the target output file is created as a new file. Backup stream data cannot be appended to an existing file. A unique file must be created for each minidisk or file space that you are backing up.

CMSFILE has no privileged access requirements beyond read/write access to the minidisk or directory that is identified in the output file specification. The calling routine is responsible for verifying that sufficient freespace is available on the output filemode to contain the resulting backup stream.

If you set **BKR_OUT_EDF_REBLOCK** to Y, and you want to configure your own UDPEs, set the following additional REXX variables for CMSFILE output:

```
BKR_OUT_EDF_UDPE1         * UDPE1 name
BKR_OUT_EDF_UDPE1_PARM    * UDPE1 parm string
BKR_OUT_EDF_UDPE1_PLEN    * UDPE1 parm length
BKR_OUT_EDF_UDPE2         * UDPE2 name
BKR_OUT_EDF_UDPE2_PARM    * UDPE2 parm string
BKR_OUT_EDF_UDPE2_PLEN    * UDPE2 parm length
```

where:

- **BKR_OUT_EDF_UDPE1**. The name of the first UDPE. By default, UDPE1 is set to BKREXT3A for CMSFILE output.
- **BKR_OUT_EDF_UDPE1_PARM**. Up to 128 bytes of parameter data for UDPE1. The default is blank. BKREXT3A does not require parameters.
- **BKR_OUT_EDF_UDPE1_PLEN**. The length of **BKR_OUT_EDF_UDPE1_PARM**.
- **BKR_OUT_EDF_UPDE2**. The second UDPE. The default is blank (disabled).
- **BKR_OUT_EDF_UPDE2_PARM**. Up to 128 bytes of parameter data for UDPE2. The default is blank.
- **BKR_OUT_EDF_UPDE2_PLEN**. The length of **BKR_OUT_EDF_UPDE2_PARM**.

Messages

Table 31 on page 78 lists the messages for CMSFILE.

Table 31. CMSFILE messages

Message number	Message text
8102	CMSFILE output handler invoked with unrecognized parameter.
8103	DMSOPEN return code *rc*, reason code *reason_code* on output file.
8104	Output handler CMSFILE initializing...
8105	Output is directed to file *file*.
8106	DMSWRITE return code *rc*, reason code *reason_code* during INIT call.
8107	DMSWRITE return code *rc*, reason code *reason_code* during FILEHEAD call.
8108	DMSWRITE return code *rc*, reason code *reason_code* during FILEDATA call.
8109	DMSWRITE return code *rc*, reason code *reason_code* during FILEEND call.
8110	DMSWRITE return code *rc*, reason code *reason_code* during EOJ call.
8111	DMSWRITE return code *rc*, reason code *reason_code* during TERMINAT call.
8112	DMSWRITE return code *rc*, reason code *reason_code* during CTNRDATA call.
8113	DMSWRITE return code *rc*, reason code *reason_code* during SFSATTR call.

DDRTAPE

The DDRTAPE input/output handler is an output-only routine that is used to create tape backups of ECKD DASD in a form that the z/VM DASD Dump Restore (DDR) utility can restore.

You cannot restore media that is produced through the DDRTAPE input/output handler using Backup and Restore Manager. Process restore operations using the CMS **DDR** command or the CP stand-alone DDR utility which can be IPLed, if needed.

DDRTAPE supports CMS-supported tape devices that also support a maximum data block size of 64K bytes. The backup data is produced in the same format as tapes that are generated by DDR using the Full Track Read (FTR) option.

You can invoke the DDRTAPE input/output handler only for output (creation of a new backup stream file) for backups of ECKD DASD minidisks or full real ECKD DASD volumes.

All tape media that are utilized by the DDRTAPE input/output handler must be pre-initialized with a valid VOL1 label using the CMS **TAPE** command with the WVOL1 parameter.

The following considerations apply to use of standard label (SL) tapes with DDRTAPE:

- The VOL1 label for the first tape that is created by a DDRTAPE backup is left intact. To restore the first output file that is created using DDRTAPE, use the SKIP 1 option of the DDR RESTORE function. (SKIP 1 causes DDR to forward space past the VOL1 label to the first output file that is created using DDRTAPE.)
- When a multiple volume set of tapes is created by a DDRTAPE backup, to maintain compatibility with the DDR output format, all volumes except for the first volume in a set, have their VOL1 label overwritten by backup output.
- Information about tapes that are used with DDRTAPE is maintained in the backup catalog in the same way as other tape-based backups.
- After a DDRTAPE backup expires, tapes that have overwritten VOL1 labels due to use in a multi-volume backup, must be re-initialized with a valid VOL1 label before Backup and Restore Manager reuses them.

DDRTAPE uses the standard CMS TAP1 virtual device address of 181 for output operations.

Because DDRTAPE handles SL tapes in a manner that differs from other tape-oriented input/output handlers, it uses a separate set of tape handling exit routines:

- BKRDRMNT. Tape mount exit routine. An alternative to the standard BKRMOUNT exit.
- BKREOV. Tape end-of-volume (EOV) exit. An alternative to the standard BKREOV exit.
- BKRDRUMT. Tape unmount exit routine. An alternative to the standard BKRUMNT exit.

The invocation syntax for each DDRTAPE exit routine is identical to their standard counterparts.

Variables required for output processing

When DDRTAPE is invoked for output processing by the DUMPCKD routine, the following variables must be present in the REXX environment:

BKR_OUT_DDR_VOLSER
A valid VOL1 identifier or the text "SCRATCH" (1-6 alphanumeric characters).

- When an explicit VOL1 label is specified, the supplied tape requires a VOL1 label that matches the specified value.
- When SCRATCH is specified, the supplied tape requires a valid VOL1 label.

The tape mount exit routine (BKRDRMNT) is responsible for verifying the SCRATCH status of the volume that is provided in response to a mount request. DDRTAPE does not inspect label contents to verify that the volume is empty or that a label-level expiration date expired.

BKR_OUT_DDR_RWSTAT
Specifies whether the system tape operator supplies media in read-only or read/write status. For DDRTAPE, always set BKR_OUT_DDR_RWSTAT to RW because DDRTAPE is used only for backup operations.

Note: For output operations, media must be mounted in read/write mode.

BKR_OUT_DDR_FILE
Specifies the forward space file (FSF) offsite to which to position the tape before the first write operation. Specify an integer greater than or equal to 1.

BKR_OUT_DDR_VERBOSE
Controls the number of console messages that are generated during processing. Specify one of the following options:

Y
Generate additional console messages during processing.

N
Suppresses non-critical messages.

BKR_DUMP_JOBSEQ
Controls whether the associated DUMP task invokes tape mount and unmount operations. Backup and Restore Manager uses when multiple DUMP routines stack output onto the target output volume. Specify one of the following options:

FIRST
Invoke the tape mount (BKRDRMNT) exit. Do not invoke the tape unmount (BKRUMNT) exit. The tape is left in position past a single tape mark, ready for the next task to begin output operations.

LAST
The tape mount exit is not invoked. The tape unmount (BKRDRUMT) exit is invoked. A double tape mark is written indicating end- of-volume. The media is rewound. The VOL1 label is checked against the value of variable **BKR_ACTUAL_DR_LABEL** to ensure that the VOL1 label was not overwritten.

ONLY
The tape mount exit (DRMNT) is invoked. The tape unmount exit (BKRDRUMT) is invoked.

INTERMED
Neither tape handling exit is invoked.

BKR_OUT_PERMIT_RETRY

Controls whether errors that have a retry option (such as a non-scratch volume that is provided in response to a scratch request, a volume label mismatch, or a read-only mount in response to a read/write mount request) is tolerated. Specify one of the following options:

Y

Recoverable mount processing errors are tolerated. The tape mount exit (BKRDRMNT) is re-driven if a recoverable error occurs during mount processing.

N

All tape mount error scenarios result in an ABEND termination.

Usage notes

Tapes that are used by DDRTAPE require a standard VOL1 label, but do not maintain extra inter-file label structures. The tape format is consistent with other standard CMS tape handling processes. Files are delimited by a single tape mark, with a double tape mark that indicates end of volume. DDRTAPE has no privileged access requirements other than provisioning of access to tape hardware and media that is consistent with the requested operation. The caller is responsible for verifying SCRATCH status of tapes that are supplied in response to a SCRATCH mount request.

Note: See the special considerations described above for information about how DDRTAPE handles SL tapes.

- If an EOV condition occurs during output, the caller must be prepared to satisfy a mount request for a SCRATCH volume to continue output. When an EOV situation is encountered, DDRTAPE drives the End-Of-Volume (EOV) exit BKRDREOV. BKRDREOV is responsible for handling label and environment variable checks, and for driving BKRDRMNT to request a SCRATCH volume for continued output.
- If an EOV condition occurs during input, the caller must be prepared to mount the appropriate successor volume in order to continue restoration of the backup stream that is being processed.

Customer applications are expected to manage their own tape cataloging requirements independently, using pre-established installation procedures or through use of a tape management system (TMS) such as IBM Tape Manager for z/VM. The supplied exit routines (BKRDRMNT, BKRDREOV, and BKRDRUMT) that are used by DDRTAPE interact with the Backup and Restore Manager catalog in a manner that is consistent with the exits (BKRMOUNT, BKREOV, BKRUMNT) that are used by other tape output handlers. The supplied DDRTAPE exit routines also interact appropriately with the backup catalog and IBM Tape Manager for z/VM if Backup and Restore Manager is configured for interaction with IBM Tape Manager for z/VM.

The DDRTAPE output handler does not support tape output reblocking or UDPE exit routines because these alternate output formats result in output that is incompatible with the z/VM DASD Dump Restore (DDR) utilities.

Note: DDRTAPE does not check the value of **BKR_Job_DDRTAPE_VERBOSE**. To receive extra console messages during DDRTAPE output processing when calling the DDRTAPE output handler directly, use the variable **BKR_OUT_DDR_VERBOSE** described in "Variables required for output processing" on page 79.

Messages

Table 32 on page 80 lists the messages for DDRTAPE.

Table 32. DDRTAPE messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8063	IJPARML format inconsistency.
8068	Invalid value for JOBSEQ: *value*.
8069	Output handler initializing with tape exit context *value*.

Table 32. DDRTAPE messages (continued)

Message number	Message text
8071	*xxx*MOUNT exit return code *rc* on primary VOL1 mount request.
8072	Return code *rc* from BKRTIO on primary GETVOL1 request.
8076	Job name is: *name*
8077	Recursive INIT call encountered.
8078	FILEHEAD called prior to initialization
8079	Recursive FILEHEAD call encountered.
8080	FHPARML inconsistency encountered.
8082	FILEDATA called prior to initialization.
8083	FILEDATA called prior to FILEHEAD.
8084	FDPARML inconsistency encountered.
8085	FILEEND called prior to initialization.
8086	FILEEND called prior to FILEHEAD.
8087	FEPARML inconsistency encountered.
8088	TERMINAT called prior to initialization.
8089	TERMINAT invoked with files still open.
8090	EOJPARML inconsistency encountered.
8093	CTNRDATA called prior to initialization.
8094	Out-of-sequence CTNRDATA call; state is not EOF.
8095	CDPARML inconsistency encountered.
8101	BKRTIO return code *rc*, reason code *reason_code*.
8114	IBMTAPE output handler invoked with unrecognized parameter.
8182	WRTAPE return code *rc*.
9144	Return code *rc* attempting to recover **BKR_ACTUAL_DR_LABEL**.
9145	Output handler DDRTAPE initializing...
9149	BKRDRUMT exit return code *rc* on dismount operation.
9159	Return code *rc* from BKRDREOV exit during end-of-volume processing.

DUALTAPE

The DUALTAPE input/output handler processes output of backup streams from a dump routine to twin IBM 34*xx* tapes with VOL1 labels. The backup stream files support a maximum data block size of 64K bytes. The backup stream files consist of a combination of client data that is contained in the backup and metadata that contains information about the task that created the backup stream.

You can invoke the DUALTAPE input/output handler for output (creation of a new backup stream file) only. The Backup and Restore Manager LOAD*xxx* routines use the IBMTAPE input/output handler where tapes were generated using DUALTAPE (because the tape format is the same). All tape media that is utilized by the DUALTAPE input/output handler must be pre-initialized with a valid VOL1 label using the CMS **TAPE** command with the WVOL1 parameter.

DUALTAPE uses the standard CMS TAP1 and TAP2 virtual device addresses of 181 and 182 for output operations.

Variables required for output processing

When DUALTAPE is invoked for output processing by the DUMPEDF, DUMPCKD, DUMPFBA, or DUMPSFS routines, the following variables must be present in the REXX environment:

BKR_OUT_PERMIT_RETRY

Controls whether errors that have a retry option (such as a non-scratch volume provided in response to a scratch request, a volume label mismatch, or a read-only mount in response to a read/write mount request) are tolerated. Specify one of the following options:

Y

Tolerate recoverable mount processing errors. The tape mount exit (BKRDRMNT) is re-driven if a recoverable error occurs during mount processing.

N

All tape mount error scenarios result in an ABEND termination.

BKR_OUT_DUAL_PRIVOL

A valid VOL1 identifier or the text "SCRATCH". Selects the primary (TAP1 / device 181) volume for a dual-tape backup.

- When an explicit VOL1 label is specified, the supplied tape requires a VOL1 label that matches the specified value.
- When "SCRATCH" is specified, the supplied tape requires a valid VOL1 label.

The client application is responsible for ensuring that the supplied volume is eligible for SCRATCH use. DUALTAPE does not inspect label contents to verify that the volume is empty or that a label-level expiration date expired. 1-8 characters.

BKR_OUT_DUAL_SECVOL

Selects the secondary (TAP2 / device 182) volume for a dual-tape backup. A valid VOL1 identifier or the text "SCRATCH".

- When an explicit VOL1 label is specified, the supplied tape requires a VOL1 label that matches the specified value.
- When "SCRATCH" is specified, the supplied tape requires a valid VOL1 label.

The client application is responsible for ensuring that the supplied volume is eligible for SCRATCH use. DUALTAPE does not inspect label contents to verify that the volume is empty or that a label-level expiration date expired. 1-8 characters.

BKR_OUT_TAPE_RWSTAT

Specifies whether the system tape operator supplies media in read-only or read/write status. For output operations, media must be mounted in read/write mode (RW).

BKR_OUT_TAPE_FILE

Specifies the forward space file (FSF) offsite to which to position the tape before the first write operations. Specify an integer greater than or equal to 1.

BKR_OUT_TAP2_FILE

Specifies the FSF offsite to which to position the tape before the first write operations. Selects the secondary (TAP2 / device 182) volume. Specify an integer greater than, or equal to 1.

BKR_OUT_TAPE_REBLOCK

Controls input/output reblocking. Specify one of the following values:

Y

Reblock data records for tape output. If you configured User Data Processing Exits (UDPEs), they are invoked as described in the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*.

N

(Default.) Do not reblock data records.

BKR_OUT_TAPE_VERBOSE

Controls the number of console messages that are generated during processing. Specify one of the following options:

Y

Generate additional console messages during processing.

N

Suppress non-critical messages.

BKR_DUMP_JOBSEQ

Controls whether the associated DUMP task invokes tape mount or unmount operations. Backup and Restore Manager uses this variable when multiple DUMP routines stack output onto the target output volume. Specify one of the following options:

FIRST

Invoke the tape mount (BKRMOUNT) exit. Do not invoke the tape unmount (BKRUMNT) exit. The tape is left in position past a single tape mark, ready for the next task to begin output operations.

LAST

Do not invoke the tape mount exit. Invoke the tape unmount (BKRUMNT) exit. A double tape mark is written indicating end- of-volume. The media is rewound. The VOL1 label is checked against the value of variable **BKR_ACTUAL_SL_LABEL** to ensure that the VOL1 label was not overwritten.

ONLY

Invoke the tape mount exit (BKRMOUNT). Invoke the tape unmount exit (BKRUMNT).

INTERMED

Do not invoke either tape handling exit.

Usage notes

Tapes that are used by DUALTAPE require a standard VOL1 label, but do not maintain extra inter-file label structures. The tape format is consistent with other standard CMS tape handling processes. Files are delimited by a single tape mark, with a double tape mark that indicates end of volume. DUALTAPE has no privileged access requirements other than provisioning of access to tape hardware and media that is consistent with the requested operation. The caller is responsible for verifying SCRATCH status of tapes that are supplied in response to a SCRATCH mount request.

DUALTAPE handling for EOV (End-of-Volume) processing differs from IBMTWIN in the following ways:

- When an IBMTWIN backup reaches EOV for either output volume, the primary and secondary volumes are closed. This behavior allows IBMTWIN to produce blockwise-identical (except for inter-file label structures) tape sets.
- When DUALTAPE encounters an EOV condition, end-of-volume processing is driven only for the volume that reached EOV. For example, if EOV is encountered on the primary (TAP1 / device 181) drive, that volume is closed and mount processing is invoked to request a new output volume for TAP1 only. This behavior is unlike IBMTWIN processing, where EOV handling is driven for both volumes when either volume reaches EOV.

  This behavior allows DUALTAPE to generate two sets of backup tapes, which viewed as a whole, are logically identical. However, every volume that is mounted for both the primary and secondary output streams are utilized to their maximum capacity. Therefore, you can use DUALTAPE to handle situations where you must generate two simultaneous copies of backup output to tape media that have dissimilar capacity.

Customer applications are expected to manage their own tape cataloging requirements independently, using pre-established installation procedures or through use of a tape management system (TMS) such as IBM Tape Manager for z/VM.

If you set **BKR_OUT_TAPE_REBLOCK** to Y, and you want to configure your own UDPEs, set the following additional REXX variables for tape output:

```
TAPE_EXIT1_NAME      * UDPE1 name
TAPE_EXIT1_PARM      * UDPE1 parm string
```

```
TAPE_EXIT1_PLEN       * UDPE1 parm length
TAPE_EXIT2_NAME       * UDPE2 name
TAPE_EXIT2_PARM       * UDPE2 parm string
TAPE_EXIT2_PLEN       * UDPE2 parm length
```

where:

- **TAPE_EXIT1_NAME**. The filename of the first UDPE. The UDPE must be available on a minidisk or directory that is accessed by the worker task service virtual machine. The configuration minidisk or directory is the recommended location.
- **TAPE_EXIT1_PARM**. Up to 128 bytes of parameter data available for UDPE1. The default is blank. BKREXT3A does not require parameters.
- **TAPE_EXIT1_PLEN**. The length of **TAPE_EXIT1_PARM**.
- **TAPE_EXIT2_NAME**. The second UDPE. The default is blank (disabled).
- **TAPE_EXIT2_PARM**. Up to 128 bytes of parameter data available for UDPE2. The default is blank.
- **TAPE_EXIT2_PLEN**. The length of **TAPE_EXIT2_PARM**.

For more information, see the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*.

Messages

Table 33 on page 84 lists the messages for DUALTAPE.

Table 33. DUALTAPE messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8065	Return code *rc* attempting to recover **BKR_ACTUAL_SL_LABEL**.
8066	Return code *rc* attempting to recover **BKR_OUT_DUAL_PRIVOL**.
8067	Return code *rc* attempting to recover **BKR_OUT_DUAL_SECVOL**.
8068	Invalid value for JOBSEQ: *value*.
8069	Output handler initializing with tape exit context *value*.
8071	*xxx*MOUNT exit return code *rc* on primary VOL1 mount request.
8072	Return code *rc* from BKRTIO on primary GETVOL1 request.
8073	*xxx*MOUNT exit return code *rc* on secondary VOL1 mount request.
8074	Return code *rc* from BKRTIO on secondary GETVOL1 request.
8075	Output handler DUALTAPE initializing...
8076	Job name is: *name*.
8077	Recursive INIT call encountered.
8078	FILEHEAD called prior to initialization.
8079	Recursive FILEHEAD call encountered.
8080	FHPARML inconsistency encountered.
8082	FILEDATA called prior to initialization
8083	FILEDATA called prior to FILEHEAD.
8084	FDPARML inconsistency encountered.
8085	FILEEND called prior to initialization.

Table 33. DUALTAPE messages (continued)

Message number	Message text
8086	FILEEND called prior to FILEHEAD.
8087	FEPARML inconsistency encountered.
8088	TERMINAT called prior to initialization.
8089	TERMINAT invoked with files still open.
8090	EOJPARML inconsistency encountered.
8091	*xxx*UMNT exit return code &1 from primary dismount.
8093	CTNRDATA called prior to initialization.
8094	Out-of-sequence CTNRDATA call; state is not EOF.
8095	CDPARML inconsistency encountered.
8096	Unrecognized CTNRDATA call type.
8097	SFSATTR called prior to initialization.
8098	SFSATTR called prior to end-of-file.
8099	SAPARML inconsistency encountered.
8100	FEOV (Forced End-of-Volume) failure for device *device*.
8101	BKRTIO return code *rc*, reason code *reason_code*.
9215	DUALTAPE output handler invoked with unrecognized parameter.
9216	Output handler DUALTAPE initializing...

IBMTAPE

Backup and restore routines use the IBMTAPE input/output handler to access backup stream content that is stored in CMS-supported tape devices, which also support a maximum data block size of 64K bytes.

The backup stream files support a maximum data block size of 64K bytes. The backup stream files consist of a combination of client data that is contained in the backup and metadata that contains information about the task that created the backup stream.

You can invoke the IBMTAPE input/output handler for output (creation of a new backup stream file) or input (access of an existing backup stream file). All tape media that is utilized by the IBMTAPE input/output handler must be pre-initialized with a valid VOL1 label. Use the CMS **TAPE** command with the WVOL1 parameter to pre-initialize tape media.

IBMTAPE uses the standard CMS TAP1 virtual device address of 181 for input and output operations.

Note: A user ID that runs *xxx*LOAD with the IBMTAPE method requires 370ACCOM to be set to ON. Each *xxx*LOAD routine issues SET 370ACCOM when needed.

Variables required for input processing

When IBMTAPE is invoked for input processing by the LOADEDF, LOADDDL, LOADFBA, LOADCKD, or LOADSFS routines, these variables must be present in the REXX environment:

BKR_INP_TAPE_VOLSER
Represents a valid VOL1 identifier. When an explicit VOL1 label is specified, the supplied tape requires a VOL1 label that matches the specified value. 1-6 characters.

BKR_INP_TAPE_RWSTAT

Specifies whether the tape is mounted (or ATTACHed by the system operator) in read-only or read/write status. Specify one of the following options:

RO

Read-only.

RW

read/write.

BKR_INP_TAPE_FILE

Specifies the FSF (forward space file) offsite to which to position the tape before the first read operation. Specify an integer greater than or equal to 1.

Variables required for output processing

When the DUMPEDF, DUMPCKD, DUMPFBA, or DUMPSFS routines invoke IBMTAPE for output processing, the following variables must be present in the REXX environment:

BKR_OUT_PERMIT_RETRY

Controls whether errors that have a retry option (such as a non-scratch volume that is provided in response to a scratch request, a volume label mismatch, or a read-only mount in response to a read/write mount request) is tolerated. Specify one of the following options:

Y

Recoverable mount processing errors are tolerated. The tape mount exit (BKRDRMNT) is re-driven if a recoverable error occurs during mount processing.

N

All tape mount error scenarios result in an ABEND termination.

BKR_OUT_TAPE_VOLSER

A valid VOL1 identifier or the word "SCRATCH".

- When an explicit VOL1 label is specified, the supplied tape requires a VOL1 label that matches the specified value.
- When SCRATCH is specified, the supplied tape requires a valid VOL1 label.

The client application is responsible for ensuring that the supplied volume is eligible for SCRATCH use. IBMTAPE does not inspect label contents to verify that the volume is empty or that a label-level expiration date expired. 1-8 characters.

BKR_OUT_TAPE_RWSTAT

Specifies whether the system tape operator supplies media in read-only or read/write status. Specify one of the following options:

RO

Read-only.

RW

read/write.

Note: For output operations, media must be mounted in read/write mode.

BKR_OUT_TAPE_FILE

Specifies the forward space file (FSF) offsite to which to position the tape before the first write operations. Specify an integer greater than or equal to 1.

BKR_OUT_TAPE_REBLOCK

Controls input/output re-blocking. Specify one of the following options:

Y

Reblock data records for tape output. If you configured User Data Processing Exits (UDPEs), they are invoked as described in the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*.

N

(Default.) Do not reblock data records.

BKR_OUT_TAPE_VERBOSE

Controls the number of console messages that are generated during processing. Specify one of the following options:

Y

Generate additional console messages during processing.

N

Suppress non-critical messages.

BKR_DUMP_JOBSEQ

Controls whether the associated DUMP task invokes tape mount or unmount operations. Backup and Restore Manager uses this variable when multiple DUMP routines stack output onto the target output volume. Specify one of the following options:

FIRST

Invoke the tape mount (BKRMOUNT) exit. Do not invoke the tape unmount (BKRUMNT) exit. The tape is left in position past a single tape mark, ready for the next task to begin output operations.

LAST

Do not invoke the tape mount exit. Invoke the tape unmount (BKRUMNT) exit. A double tape mark is written to indicate end- of-volume. The media is rewound. The VOL1 label is checked against the value of variable **BKR_ACTUAL_SL_LABEL** to ensure that the VOL1 label was not overwritten.

ONLY

Invoke the tape mount exit (BKRMOUNT). Invoke the tape unmount exit (BKRUMNT).

INTERMED

Do not invoke either tape handling exit.

Usage notes

Tapes that are used by IBMTAPE require a standard VOL1 label, but do not maintain additional inter-file label structures. The tape format is consistent with other standard CMS tape handling processes. Files are delimited by a single tape mark, with a double tape mark that indicates end of volume. IBMTAPE has no privileged access requirements other than provisioning of access to tape hardware and media that is consistent with the requested operation. The caller is responsible for verifying SCRATCH status of tapes that are supplied in response to a SCRATCH mount request.

- If an EOV condition occurs during output, the caller must be prepared to satisfy a mount request for a SCRATCH volume to continue output.
- If an EOV condition occurs during input, the caller must be prepared to mount the appropriate successor volume to continue restoration of the backup stream being processed.

Customer applications are expected to manage their own tape cataloging requirements independently, using pre-established installation procedures or through use of a tape management system (TMS) such as IBM Tape Manager for z/VM.

If you set **BKR_OUT_TAPE_REBLOCK** to Y, and you want to configure your own UDPEs, set the following additional REXX variables for tape output:

```
TAPE_EXIT1_NAME      * UDPE1 name
TAPE_EXIT1_PARM      * UDPE1 parm string
TAPE_EXIT1_PLEN      * UDPE1 parm length
TAPE_EXIT2_NAME      * UDPE2 name
TAPE_EXIT2_PARM      * UDPE2 parm string
TAPE_EXIT2_PLEN      * UDPE2 parm length
```

where:

- **TAPE_EXIT1_NAME**. The name of the first UDPE. The UDPE must be available on a minidisk or directory that is accessed by the worker task service virtual machine. The configuration minidisk or directory is the recommended location.

- **TAPE_EXIT1_PARM**. Up to 128 bytes of parameter data available for UDPE1. The default is blank. BKREXT3A does not require parameters.
- **TAPE_EXIT1_PLEN**. The length of **TAPE_EXIT1_PARM**.
- **TAPE_EXIT2_NAME**. The second UDPE. The default is blank (disabled).
- **TAPE_EXIT2_PARM**. Up to 128 bytes of parameter data available for UDPE2. The default is blank.
- **TAPE_EXIT2_PLEN**. The length of **TAPE_EXIT2_PARM**.

For more information, see the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*.

Messages

Table 34 on page 88 lists the messages for IBMTAPE.

Table 34. IBMTAPE messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8065	Return code *rc* attempting to recover **BKR_ACTUAL_SL_LABEL**.
8068	Invalid value for JOBSEQ: *value*.
8069	Output handler initializing with tape exit context *value*.
8071	*xxx*MOUNT exit return code *rc* on primary VOL1 mount request.
8072	Return code *rc* from BKRTIO on primary GETVOL1 request.
8076	Job name is: *name*.
8077	Recursive INIT call encountered.
8078	FILEHEAD called prior to initialization.
8079	Recursive FILEHEAD call encountered.
8080	FHPARML inconsistency encountered.
8082	FILEDATA called prior to initialization.
8083	FILEDATA called prior to FILEHEAD.
8084	FDPARML inconsistency encountered.
8085	FILEEND called prior to initialization.
8086	FILEEND called prior to FILEHEAD.
8087	FEPARML inconsistency encountered.
8088	TERMINAT called prior to initialization.
8089	TERMINAT invoked with files still open.
8090	EOJPARML inconsistency encountered.
8091	*xxx*UMNT exit return code *rc* from primary dismount.
8093	CTNRDATA called prior to initialization.
8094	Out-of-sequence CTNRDATA call; state is not EOF.
8095	CDPARML inconsistency encountered.
8096	Unrecognized CTNRDATA call type.

Table 34. IBMTAPE messages (continued)

Message number	Message text
8097	SFSATTR called prior to initialization.
8098	SFSATTR called prior to end-of-file.
8099	SAPARML inconsistency encountered.
8101	BKRTIO return code *rc*, reason code *reason_code*.
8114	IBMTAPE output handler invoked with unrecognized parameter.
8115	Output handler IBMTAPE initializing...
8216	GETDATA call received prior to initialization.
8217	BKRTIO return code *rc* while reading input stream.

IBMTWIN

The IBMTWIN input/output handler processes output of backup streams from a dump routine to twin IBM 34*xx* tapes with VOL1 labels. The backup stream files support a maximum data block size of 64K bytes and consist of a combination of client data that is contained in the backup, as well as metadata that contains information about the task that created the backup stream.

Invoke the IBMTWIN input/output handler for output (creation of a new backup stream file) only. The Backup and Restore Manager LOAD*xxx* routines use the IBMTAPE input/output handler where tapes were generated through IBMTWIN (because the tape format is the same). All tape media that is utilized by the IBMTWIN input/output handler must be pre-initialized with a valid VOL1 label using the CMS **TAPE** command with the WVOL1 parameter.

IBMTAPE uses the standard CMS TAP1 virtual device address of 181 for output operations.

Variables required for output processing

When the DUMPEDF, DUMPCKD, DUMPFBA, or DUMPSFS routines invoke IBMTWIN for output processing, the following variables must be present in the REXX environment:

BKR_OUT_PERMIT_RETRY

Controls whether errors that have a retry option (such as a non-scratch volume that is provided in response to a scratch request, a volume label mismatch, or a read-only mount in response to a read/write mount request) are tolerated. Specify one of the following options:

Y

Tolerate recoverable mount processing errors. The tape mount exit (BKRDRMNT) is re-driven if a recoverable error occurs during mount processing.

N

All tape mount error scenarios result in an ABEND termination.

BKR_OUT_TAPE_VOLSER

A valid VOL1 identifier or the text "SCRATCH".

- When an explicit VOL1 label is specified, the supplied tape requires a VOL1 label that matches the specified value.
- When "SCRATCH" is specified, the supplied tape requires a valid VOL1 label.

The client application is responsible for ensuring that the supplied volume is eligible for SCRATCH use. IBMTWIN does not inspect label contents to verify that the volume is empty or that a label-level expiration date expired. 1-8 characters.

BKR_OUT_TAPE_RWSTAT
Specifies whether the system tape operator supplies media in read-only or read/write status. For output operations, media must be mounted in read/write mode (RW).

BKR_OUT_TAPE_FILE
Specifies the forward space file (FSF) offsite to which to position the tape before the first write operations. Specify an integer greater than or equal to 1.

BKR_OUT_TAPE_CURVOL
The current VOL1 label at termination time.

BKR_OUT_TAPE_NXTFIL
The current FSF offset for the "next" file at termination time (for example, the "next file" offset).

BKR_OUT_TAPE_REBLOCK
Controls input/output reblocking. Specify one of the following options:

Y
Re-block data records for tape output. If you configured UDPEs (User Data Processing Exits), they are invoked as described in the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*.

N
(Default.) Do not re-block records.

BKR_OUT_TAPE_VERBOSE
Controls the number of console messages that are generated during processing. Specify one of the following options:

Y
Generate additional console messages during processing.

N
Suppress non-critical messages.

BKR_DUMP_JOBSEQ
Controls whether the associated DUMP task invokes tape mount or unmount operations. Backup and Restore Manager uses this variable when multiple DUMP routines stack output onto the target output volume. Specify one of the following options:

FIRST
Invoke the tape mount (BKRMOUNT) exit. Do not invoke the tape unmount (BKRUMNT) exit. The tape is left in position past a single tape mark, ready for the next task to begin output operations.

LAST
Do not invoke the tape mount exit. Invoke the tape unmount (BKRUMNT) exit. A double tape mark is written to indicate end- of-volume. The media is rewound. To ensure that the VOL1 label was not overwritten, the VOL1 label is checked against the value of variable **BKR_ACTUAL_SL_LABEL**.

ONLY
Invoke the tape mount exit (BKRMOUNT). Invoke the tape unmount exit (BKRUMNT).

INTERMED
Do not invoke either tape handling exit.

Usage notes

Tapes that are used by IBMTWIN require a standard VOL1 label, but do not maintain additional inter-file label structures. The tape format is consistent with other standard CMS tape handling processes. Files are delimited by a single tape mark with a double tape mark that indicates end of volume. IBMTWIN has no privileged access requirements other than provisioning of access to tape hardware and media that is consistent with the requested operation. The caller is responsible for verifying SCRATCH status of tapes that are supplied in response to a SCRATCH mount request.

- If an EOV condition occurs during output, the caller must be prepared to satisfy a mount request for a SCRATCH volume to continue output.
- An EOT condition on either output tape drive triggers EOT processing on both output drives.

Customer applications are expected to manage their own tape cataloging requirements independently, either using pre-established installation procedures or through use of a tape management system (TMS) such as IBM Tape Manager for z/VM.

If you set **BKR_OUT_TAPE_REBLOCK** to Y, and you want to configure your own UDPes, set the following additional REXX variables for tape output:

```
TAPE_EXIT1_NAME      * UDPE1 name
TAPE_EXIT1_PARM      * UDPE1 parm string
TAPE_EXIT1_PLEN      * UDPE1 parm length
TAPE_EXIT2_NAME      * UDPE2 name
TAPE_EXIT2_PARM      * UDPE2 parm string
TAPE_EXIT2_PLEN      * UDPE2 parm length
```

where:

- **TAPE_EXIT1_NAME**. The filename of the first UDPE. The UDPE must be available on a minidisk or directory that is accessed by the worker task service virtual machine. The configuration minidisk or directory is the recommended location.
- **TAPE_EXIT1_PARM**. Up to 128 bytes of parameter data available for UDPE1. The default is blank. BKREXT3A does not require parameters.
- **TAPE_EXIT1_PLEN**. The length of **TAPE_EXIT1_PARM**.
- **TAPE_EXIT2_NAME**. The second UDPE. The default is blank (disabled).
- **TAPE_EXIT2_PARM**. Up to 128 bytes of parameter data available for UDPE2. The default is blank.
- **TAPE_EXIT2_PLEN**. The length of **TAPE_EXIT2_PARM**.

For more information, see the *IBM Backup and Restore Manager for z/VM Administration Guide (SC18-9346)*.

Messages

Table 35 on page 91 lists the messages for IBMTWIN.

Table 35. IBMTWIN messages

Message number	Message text
8001	BKRGETRX return code *rc* while trying to fetch *variable*.
8065	Return code *rc* attempting to recover **BKR_ACTUAL_SL_LABEL**.
8066	Return code *rc* attempting to recover **BKR_OUT_TWIN_PRIVOL**.
8067	Return code *rc* attempting to recover **BKR_OUT_TWIN_SECVOL**.
8068	Invalid value for JOBSEQ: *value*.
8069	Output handler initializing with tape exit context *value*.
8071	*xxx*MOUNT exit return code *rc* on primary VOL1 mount request.
8072	Return code *rc* from BKRTIO on primary GETVOL1 request.
8073	*xxx*MOUNT exit return code *rc* on secondary VOL1 mount request.
8074	Return code *rc* from BKRTIO on secondary GETVOL1 request.
8075	Output handler IBMTWIN initializing...
8076	Job name is: *name*.
8077	Recursive INIT call encountered.
8078	FILEHEAD called prior to initialization.
8079	Recursive FILEHEAD call encountered.

Table 35. IBMTWIN messages (continued)

Message number	**Message text**
8080	FHPARML inconsistency encountered.
8082	FILEDATA called prior to initialization.
8083	FILEDATA called prior to FILEHEAD.
8084	FDPARML inconsistency encountered.
8085	FILEEND called prior to initialization.
8086	FILEEND called prior to FILEHEAD.
8087	FEPARML inconsistency encountered.
8088	TERMINAT called prior to initialization.
8089	TERMINAT invoked with files still open.
8090	EOJPARML inconsistency encountered.
8091	*xxx*UMNT exit return code *rc* from primary dismount.
8093	CTNRDATA called prior to initialization.
8094	Out-of-sequence CTNRDATA call; state is not EOF.
8095	CDPARML inconsistency encountered.
8096	Unrecognized CTNRDATA call type.
8097	SFSATTR called prior to initialization.
8098	SFSATTR called prior to end-of-file.
8099	SAPARML inconsistency encountered.
8100	FEOV (Forced End-of-Volume) failure for device *device*.
8101	BKRTIO return code *rc*, reason code *reason_code*.

Notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119

Armonk, NY 10504-1785
US.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Index

Product Number: 5697-J06

SC18-9523-15